Oriented patterns in image analysis

Hugues Talbot

ETUDE DES DIRECTIONS EN ANALYSE D'IMAGE

Des objets fins aux méthodes à base de flot

THÈSE D'HABILITATION

présentée à
l'Université Paris-Est
par

Hugues TALBOT

pour l'obtention de

L'HABILITATION
à
DIRIGER DES RECHERCHES

Version 0.99 du 9 décembre 2013

Cette thèse sera soutenue le 13 decembre 2013 à Noisy-le-Grand
devant le jury composé de :

Isabelle	Bloch	*Rapporteur*
Daniel	Cremers	*Rapporteur*
François	Malgouyres	*Rapporteur*
Gilles	Bertrand	*Examinateur*
Jocelyn	Chanussot	*Examinateur*
Abderrahim	Elmoataz	*Examinateur*
Grégoire	Malandain	*Examinateur*
Jean	Serra	*Examinateur*
Olga	Veksler	*Examinateur*

Think globally, act locally [1].

À la mémoire de mes grands-parents Juliette & Marcel Talbot

[1] Attributed to town planner and social activist Patrick Geddes, 1915

Thanks

I would like to most sincerely thank the "rapporteurs" of this thesis: Isabelle Bloch, Daniel Cremers and François Malgouyres, who have kindly accepted to take part in this jury, to read this manuscript carefully and to produce a review of its content. Also, I would like to thank all the members of the jury, so in addition to the rapporteurs: Gilles Bertrand, Jocelyn Chanussot, Abderrahim Elmoataz, Grégoire Malandin, Jean Serra and Olga Veksler, who have kindly accepted to donate significant time and effort to this exercise.

Thanks also to the scientific and administrative staff who made this thesis and associated events possible: Latifa Zeroual-Belou, Stéphane Bouton, Sylvie Cach, Martine Elichabe, and the Conseil Scientifique de l'Université Paris Est.

Prelude

The work presented in this thesis extends from 1987 to 2013, a unusually long period for an "habilitation" thesis. Significantly, they took place in three different countries: Australia, France and the USA. A lot of people contributed to this work. It is unfortunate that I will most likely forget to thank all of those who would deserve a mention. Tens of pages would not be sufficient. I hope I can be forgiven.

I will nonetheless attempt to go through a list, in chronological order, which will allow me to also outline a personal and scientific path.

First of all there are moments in life that change if not whole lives, at least careers. One of these moments happened when I was hailed one day in November 1987 by my friend Denis Leconte, a.k.a "Nop", then intern at the AAI company, operating at the Ecole Centrale de Paris in the electronics lab, to see if I was interested in an internship too. It would involve a project: implementing some image analysis operators from a weird theory called Mathematical Morphology. I said yes, of course. Denis, one year my senior, had already been a mentor at ECP in various technical matters, and I trusted him completely. Denis later emigrated to the USA, worked at JPL, married, had kids there and recently got a technical Oscar for his imaging work in Hollywood.

Another interesting employee of AAI, beginning her thesis was Isabelle Bloch. She gave me the first explanation of what were erosions, dilations, distance transforms, skeletization and so on; she gave me many articles to read and provided all the help I needed. This collaboration with AAI lasted two years, and we never really lost touch with one another following this.

A little bit later, in 1989, three persons from Saint-Gobain: Jean-Pierre Poitevin, Catherine Langlais and Daniel Hanton read my CV, found the line concerning image analysis and mathematical morphology and on this basis offered me a "Volontariat du Service National Actif" (VSNA) at MIT to work on the segmentation and measurement of man-made vitreous fibres, as well as a CIFRE convention to start a PhD, in lieu of a more standard military service.

Of course this sentence is a huge shortcut. A host laboratory was needed, for this none other than the Centre de Morphology Mathématique, at the Ecole des Mines de Paris, then headed by Jean Serra, agreed to take me on. At the time an engineering diploma was not sufficient to start a Doctorat, so Jean-Claude Simon agreed to take me in his DEA, the IARFAG of then Université Paris-VI, even as courses were about to start.

In Cambridge, MA, Linn W. Hobbs, professor in the materials science department at MIT, agreed to accept me in his lab. I was the sole computer scientist among all these physicists, but I was wholly welcome (probably because I could maintain the VAX computer, but still...). I spent a very productive and enjoyable time at MIT between January 1991, the start of the first gulf war, and October 1992. I used a NeXT workstation purchased by Saint-Gobain, studying the samples imaged in the experimental electron microscopes of the department, all

the while making many friends, including Lu-Chang and Shreeram among them.

Back from the USA, thanks to Dominique Jeulin, I finished my PhD dissertation under the best conditions, since Jean Serra was in Barcelona on a sabbatical. At the end of 1993 I defended my thesis in the presence, among others, of all three of my directors. This period had been extremely enjoyable with the start of long lasting friendships with Marc Van Droogenbroeck, Pierre Soille, Corinne Vachier, Luc Vincent and almost daily interactions with other mentors at the CMM including Serge Beucher, Michel Bilodeau, Jean-Claude Klein and Fernand Meyer. This was also a period where I was very lucky to meet people who were very influential for me, including Georges Matheron, Michel Schmitt, Petros Maragos, Gaile Gordon who was doing her thesis with David Mumford and who married Luc Vincent. I also remember fondly Fred Blundell, a mathematician who asked me once to explain to him mathematical morphology, and Carl Feynman, son of Richard, who was working at the short-lived but influential Thinking Machine Inc.

After my doctorate, I worked for a year at Saint Gobain, implementing and optimizing the results of my thesis. However I was now destined to forever be a willing victim to the siren calls of scientific research. French Premier Balladur cut national research funding in 1994, which together with high unemployment rates, prompted an exile to Australia.

I was again welcomed with open arms Down Under, again by exceptional people: Murray Cameron, who later became the Chief of the Mathematical and Information Science division of CSIRO, and Mark Berman, a man of impeccable scientific and human credentials, and absolute integrity. There I had extremely productive interactions with my direct colleagues: Richard Beare, Leanne Bischof, Michael Buckley, Ronald Jones, Changming Sun, later Paul Jackway and Sébastien Ourselin. In these ten years in Australia, I had quite varied experiences, from the early enthusiasm of the melanoma project with the Polartechnics company and Scott Menzies of the Sydney Melanoma Unit at Royal Prince Alfred Hospital, to the technical and financial success of the Axon automated microscope almost all the way to the end. In Australia I was honored to work a little with visitors James Sethian, Christian Ronse and Alexandre Tuzikov. A very special mention must be given to Henk Heijmans, whose cerebral vascular accident will never leave my memory. I experienced the amazing dedication and bigger than life persona of Mervyn Thomas, as well as the less enjoyable aspects of working under the constant financial pressure of industrial research. I witnessed both the best and the worst practices of management. The latter lead to my family's eventual decision to come back to France, family that had been extended with the arrival of a little Australian girl four years earlier.

In 2004, taking advantage of a sabbatical year, I was welcomed in an excellent team, which was and is still being lead by Gilles Bertrand, and that include now very close colleagues: Mohamed Akil, Tarik Al Ani, Christine Auger, Alex Hamam, Lilian Buzer, Michel Couprie, Christophe Dietrich, Thierry Grandpierre, Yukiko Kenmochi, Eric Llorens, Laurent Najman, Laurent Perroton, François Rocariès and Jérôme Sueur. Over the years the team has expanded, and now includes former student Jean Cousty, as well as Eva Dokladalova, Nabil Mustapha, Frank Schmidt and indefatigable and more active than ever Jean Serra, not to mention our latest colleague Benjamin Perret. Of course I must also mention Dror Aiger. In spite of his short tenure with us we managed to write together five articles. Thanks to these stimulating surroundings I have had the fantastic luck to be able to interact and collaborate with great scientists and people like Dominique Bernard, Caroline Chaux, Emilie Chouzenoux, Théo Géraud, Emmanuelle Gouillart, Amir Nakib, Nikos Paragios, Jean-Christophe Pesquet, Nelly Pustelnik as well as visitors Yuri Boykov, Leo Grady and Olga Veksler, to name but a few.

I should have finished this thesis much much much earlier, since perhaps the material was not lacking, but there was always seemingly better things to do: extend the family some more, work on articles and books, help students, and just do research with the fantastic people around me.

I would like to finish by thanking some very important people: the PhD student I have helped supervise: Ben Appleton, Fiona Evans, Harold Phelippeau, Olena Tankyevych, Camille Couprie, László Marak, Nicolas Combaret, Ania Jezierska, Ngo Phuc, Imen Melki, Eloïse Grossiord, Odyssée Merveille and Ali Kanj. They probably didn't know what to expect when they started, I hope they will have received at least some measure of what they gave to me. I would also like to thank the PhD students with whom I have worked with great pleasure even though I was not their official co-supervisor: Jean Cousty, Yohann Thibaut, John Chaussard, Emilie Charrier, Roland Levillain, David Menotti, Vincent Bismuth and Elodie Puybareau, as well as those not directly inside our research team, but with whom I have had a lot of interaction: Erwan Plougonven, Silvia Valero and François Cokelaer.

Let me thank one more time the readers of this thesis, and again each of the members of the jury, especially the rapporteurs.

Last, but of course not least, I would like to thank my daughters Zoé and Sophie and finally my most excellent wife Annick, amie, compagne et puis épouse, témoin et soutien indéfectible de tous ces évènements. Sans toi je n'aurais rien pu faire.

Contents

Table of contents **1**

General introduction **7**
Thin objects . . . 7
Flow methods in image analysis . . . 8
Objective and outline of the thesis . . . 8

I Thin objects: problems, review, filtering and segmentation **9**

Introduction **11**

1 Thin objects in imaging **13**
1.1 Examples of problems that include thin objects . . . 13
1.1.1 Definitions and recognition . . . 13
1.1.2 Definition of thin objects . . . 15
1.1.3 Orientation and scale . . . 17
1.2 Application fields . . . 17
1.2.1 Some 2D problems . . . 17
1.2.1.1 Neurites . . . 18
1.2.1.2 Retina blood vessels . . . 19
1.2.1.3 X-ray images of stent guide . . . 20
1.2.1.4 Oriented textures . . . 20
1.2.1.5 Oriented elements in photography . . . 21
1.2.2 2D+t problems . . . 22
1.2.3 Tracking small compact objects in time . . . 23
1.2.4 Injected X-ray angiography . . . 23
1.2.5 3D problems . . . 24
1.2.5.1 Fibres in 3D . . . 24
1.2.5.2 Vessels in medical imaging . . . 25
1.2.5.3 Cracks in concrete . . . 26
1.2.5.4 Seismic data, fracture lines . . . 26
1.2.6 3D+t problems . . . 26
1.2.6.1 Cracks in concrete over time . . . 26
1.3 Summary . . . 28

2 Thin object filtering and segmentation **31**
2.1 Linear filtering . . . 31
2.1.1 Local derivative-based oriented filters . . . 31
2.1.1.1 The gradient operator . . . 31
2.1.1.2 The Hessian operator . . . 32
2.1.1.3 Multi-resolution - Scale-space - Vesselness functions . . . 33
2.1.1.4 Structure tensor . . . 34
2.1.2 Filter banks . . . 35
2.1.2.1 Steerable filters . . . 35
2.1.2.2 Directional filter banks . . . 37
2.1.3 Orientation-space . . . 39
2.1.4 Wavelets . . . 39
2.1.4.1 Contourlets . . . 39
2.1.4.2 Ridgelets . . . 41
2.1.4.3 Curvelets . . . 43
2.1.4.4 Bandelets . . . 44
2.1.4.5 Other Xlets . . . 45
2.1.5 Non-local filtering and patches . . . 45
2.2 Non-linear, morphological operators . . . 47
2.2.1 Operations with line segments as structuring elements . . . 47
2.2.2 Binary-only approaches . . . 49
2.2.3 Local minimal path . . . 49
2.2.4 Connected morphological filters . . . 50
2.2.5 Geodesic paths and path voting . . . 50
2.2.6 Other approaches . . . 51
2.2.7 Demosaicing . . . 51
2.2.8 Discussion . . . 51
2.3 Contributions to thin objects filtering . . . 52
2.3.1 Segment-based operators . . . 52
2.3.1.1 Faster, less noise sensitive, and translation-invariant morphological operators with line segments . . . 52
2.3.1.2 Restoring the translation invariance . . . 53
2.3.1.3 Improving noise sensitivity . . . 55
2.3.1.4 Other remarks . . . 56
2.3.2 Operations with paths as structuring elements . . . 57
2.3.2.1 Graphs and path operators . . . 57
2.3.2.2 Path-based operators and noise sensitivity . . . 59
2.3.2.3 Applications in 2D . . . 61
2.3.2.4 Extensions to 3D . . . 61
2.3.3 Other approaches . . . 63
2.4 Conclusion . . . 63

II Flow methods in image analysis and computer vision **67**

Introduction **69**

3 Seeded segmentation methods **71**
3.1 The need for seed-driven segmentation . . . 71
3.1.1 Image analysis and computer vision . . . 72
3.1.2 Objects are semantically consistent . . . 72
3.1.3 A separation of powers . . . 73
3.1.4 Desirable properties of seeded segmentation methods . . . 74
3.2 A review of segmentation techniques . . . 74
3.2.1 Pixel selection . . . 75
3.2.2 Contour tracking . . . 75
3.2.3 Statistical methods . . . 76
3.2.4 Continuous optimization methods . . . 77
3.2.4.1 Active contours . . . 77
3.2.4.2 Level sets . . . 78
3.2.4.3 Geodesic active contours . . . 79
3.2.5 Graph-based methods . . . 80
3.2.5.1 Graph cuts . . . 81
3.2.5.2 Random walkers . . . 82
3.2.5.3 Watershed . . . 83
3.2.6 Generic Models for Segmentation . . . 85
3.2.6.1 Continuous models . . . 86
3.2.6.2 Hierarchical models . . . 86
3.2.6.3 Convex and convex relaxation methods . . . 87
3.2.6.4 Combinations . . . 87
3.3 A unifying framework for discrete seeded segmentation . . . 87
3.3.1 Discrete optimization . . . 88
3.3.2 A unifying framework . . . 88
3.3.3 Power watershed . . . 89
3.4 Globally optimum continuous segmentation methods . . . 91
3.4.1 Dealing with noise and artifacts . . . 91
3.4.2 Globally optimal Geodesic Active Contour . . . 92
3.4.3 Maximal continuous flows and Total Variation . . . 92
3.5 Comparison and discussion . . . 92
3.6 Conclusion and future work . . . 95

4 Discrete Flow-based methods **97**
4.1 Discrete flows in networks . . . 97
4.1.1 Motivation . . . 97
4.1.2 Flows in graphs . . . 97
4.1.3 Minimum cost graph cuts . . . 98
4.1.4 Images as graphs . . . 99
4.2 The Ford-Fulkerson maxflow-mincut theorem . . . 99
4.3 Examples and remarks . . . 100
4.4 Augmenting path algorithms . . . 101
4.5 Push-relabel algorithms . . . 102
4.6 Classical algorithms and recent advances . . . 104
4.7 Applications . . . 105
4.7.1 Seeded image segmentation . . . 105

4.7.1.1 Segmentation using unary weights: . . . 106
4.7.2 Maximum A-Posteriori - Markov Random Fields interpretation . . . 107
4.8 A general discrete framework for energy minimization . . . 109
4.8.1 The Ishikawa construction . . . 110
4.8.2 Image restoration using the Ishikawa graph . . . 110
4.9 The move framework . . . 113
4.9.1 Binary representation and sub-modularity . . . 114
4.9.1.1 Sub-modularity . . . 114
4.9.2 The expansion move . . . 115
4.9.2.1 Principle and convergence of expansion moves . . . 117
4.9.3 Convex and Non-convex moves . . . 118
4.9.3.1 Joint quantization and denoising . . . 118
4.9.3.2 Truncated convex moves . . . 118
4.10 Power watershed . . . 120
4.10.1 Image filtering . . . 122
4.10.2 Surface reconstruction from isolated points . . . 125
4.11 Conclusion and future work . . . 126

5 Continuous Flow-based methods 129
5.1 Motivation . . . 129
5.2 Previous work . . . 130
5.2.1 Functions with bounded variation . . . 130
5.2.2 Iri's work . . . 131
5.2.3 Strang's work on maximum flow through a continuous domain . . . 132
5.2.3.1 Maximum flow through a domain . . . 133
5.2.3.2 Minimum cut in a continuous domain . . . 133
5.2.3.3 Duality and Total Variation . . . 134
5.2.3.4 Condition for optimality . . . 135
5.2.3.5 Existence of a minimizing series of characteristic functions . . . 136
5.2.4 The Nozawa conditions . . . 136
5.2.5 Total variation minimization algorithms . . . 137
5.2.5.1 Chambolle's algorithm . . . 137
5.2.6 Proximal methods . . . 140
5.3 Continuous maximum flows . . . 140
5.3.1 A continuous flow simulation . . . 140
5.3.2 Relation to TV minimization . . . 141
5.3.3 Generalization to primal-dual convex optimization . . . 143
5.4 Convex optimization on graphs . . . 145
5.4.1 Continuous maximum flows on graphs . . . 145
5.4.1.1 Graph-based formulation of the CMF problem . . . 145
5.4.1.2 Dual formulation . . . 146
5.4.1.3 Solving the CCMF problem . . . 148
5.4.1.4 Results . . . 148
5.4.2 Dual-constrained regularization on graphs . . . 149
5.4.2.1 Example of interesting class of constraint sets . . . 151
5.4.2.2 Proposed algorithms for DCTV . . . 152
5.4.2.3 Results . . . 154

5.5 Extensions . . . 157
5.6 Conclusion and future work . . . 158

6 General conclusion and future work 161
6.1 Past work . . . 161
6.2 Current planned work . . . 162
6.2.1 Segmentation and medical imaging . . . 162
6.2.2 Joint segmentation, reconstruction and restoration . . . 162
6.2.3 Discrete transforms . . . 162
6.2.4 Vivabrain . . . 162
6.3 Unplanned future work . . . 163

III Appendix 165

Content of the appendix 167

A CV of Hugues Talbot 169

B Publications list 175

C Industrial contributions 187
C.1 Axon Instruments imageXpress 5000 . . . 187
C.2 Polartechnics Solarscan . . . 195

D Directional Morphological Filtering 203

E Path Openings and Closings 223

F Joint Quantizing and Denoising 239

G Continuous Maximum Flows 257

H Power Watersheds 273

LICENSE 291

Table des figures 292

Liste des tableaux 296

Bibliography 297

Index 317

General introduction

The work presented in this habilitation thesis deals with several topics, related to aspects of image analysis and image processing I have contributed to. There are essentially two parts to this document: the first part deals with the filtering and segmentation of thin objects in 2D and 3D images ; the second part concerns optimal discrete and continuous method for segmentation based on the notion of flow. There is a link between these two topics, which will be provided in the course of the thesis.

Thin objects

Thin objects are prevalent in images. For example, one can think of the following:

- Fibrous objects in materials, e.g. composite with glass or carbon fibres;
- fracture lines and surfaces;
- thin anatomical structures: blood vessels, muscle fibres, neurites;
- some oriented textures in clothes or hair.

The list of potential examples is nearly infinite. There does not exist an unambiguous definition of what constitutes a thin object, but for the purpose of explanations, we will say that an object is thin if it semantically coherent and at least one of its dimensions is much smaller than the others. Contrariwise, we will say that an object is "isotropic" if it is non-thin, i.e. all of its dimensions are comparable.

In the general literature on image processing and analysis, thin objects are not usually mentioned as deserving special treatment. However, because they possess this dimension that is much smaller than the others, they are indeed often harder to acquire, process, segment and analyse than more "isotropic" objects. For image filtering, for instance, one often uses masks or windows of some fixed dimension like 3×3 or 5×5 pixels. Usually, one makes the assumption that these masks entirely fits into most of the objects of interest. This may not be true for a thin object. In segmentation, many popular methods assume that one can start from some starting point in an object, and propagate information around this starting point until the contours of the object are found. For thin objects, one may not be able to perform this propagation, and moreover, one may not even be able to define their contours!

Thin objects are not isotropic, they are usually elongated in one or more direction and/or dimension and so locally oriented. Discovering, measuring and using this information is important in a number of applications. This makes is possible for instance to characterize some textures. Measuring anisotropy and orientation has several applications in materials science.

A single object in image analysis are often assumed to be connected. On images, due to noise and discretization issues, thin objects might be locally disconnected. Whereas a human observer might still recognize semantically a hair, a fibre or a vessel, at the local pixel level, the information is often lost in the noise. How then is it possible to connect high-level information to low-level vision operators? This is a prevalent problem in computer vision and image analysis, but is it particularly difficult to solve for thin objects.

In the first part of thesis, we will explore ways to deal with thin objects, particularly using the tools of mathematical morphology, but also optimization-based approaches. A thin object in 2D or 3D can be detected using minimal paths approaches. This leads us to the second part of this thesis.

Flow methods in image analysis

Optimization-based approaches for finding thin objects have also been used in other contexts. Indeed, the contour of an isotropic, non-thin object is also a thin object. For instance, minimal path and minimal surfaces computations can be performed in the dual. One well-known discrete method in this context is the graph maxflow-mincut theorem of Ford and Fulkerson. This has led us to consider the problem of object segmentation from an optimization perspective, and to develop new tools in this context.

Image segmentation, i.e. the act (or art) of delineating objects in images is an essential step of many applications. As research progressed over the years, segmentation methods have simultaneously moved from simple pixel-based techniques, to statistical characterization, to regions-based methods. At the end of the 1980s, segmentation was finally seen as a variational problem with associated energy minimization schemes. This viewpoint enabled the use of iterative contour and surface evolution techniques of increased sophistication. Around the year 2000, techniques for finding the global optimum of some of these formulations became finally available.

In this part of the thesis, we will describe some of these techniques we have contributed to. We will also describe image processing techniques for denoising or reconstruction that are related to this effort.

Objective and outline of the thesis

The objective of this thesis is not to describe the solution to the problems described above, it is more modestly to propose a context for these study, and to illustrate the following points:

- What is the problem we are trying to solve ;
- What are the main tools that exist or were developed ;
- What was our contribution to these tools.

This is done via a main text, followed by some selected publications. In the course of writing this thesis, I have sought to present topics in a didactic manner in the main body of the text. I have in places spent more space in the main text to the work of others instead of my own, as an introduction to reading and understanding the articles that follow. It is my hope that it will be useful.

Part I

Thin objects: problems, review, filtering and segmentation

Introduction

In this part, we will introduce the general problem of thin object filtering and segmentation. We will start from illustrative example in 2D, 2D+t and 3D. In a second chapter, we will propose a literature review of the topic. We will then propose filtering and segmentation method for these object, together with qualitative and quantitative assessment of their performance. In a last chapter for this part, we will introduce extensions, future work and a link to the next part.

Chapter 1

Thin objects in imaging

In this introductory chapter, we present the general framework of this study, in particular, using illustrative examples, we present various problems associated with thin objects.

1.1 Examples of problems that include thin objects

Thin objects are in fact common in imaging problems, but do not appear so significantly when studying image analysis, or in the bulk of publications related to that topic.

In this section, we show some examples of images that include thin objects, and the problems associated with them. We start with some concepts and definitions.

1.1.1 Definitions and recognition

Segmentation is act (or the art) of delineating objects of interests in images. This definition is more semantic than mathematical because at its heart so is segmentation itself. In fact segmentation is essentially a cognitive concept because an object generally cannot be defined precisely. Where is the contour of a cloud? of a tree? of a complex organ like the lung? In the medical imaging context, even experts often do not agree on the placement of contours of interest.

The classical approach to segmentation, sometimes called *top-down* (Beucher, 1990) implies to segment an image first, in order to simplify its content. According to this view, segmentation is therefore a kind of filtering operator. Object recognition comes only after, perhaps after grouping or splitting already segmented regions. In contrast, the so-called morphological vision of segmentation, sometimes called *bottom-up* and championed by Meyer and Beucher (1990) supposes we know in advance the characteristics of the objects we would like to segment. Although this may sound like a catch-22 situation, this is not really the case. In the case of image analysis (say for the diagnosis of skin lesions or the counting of red blood cell in a smear), we do largely know in advance what we are going to observe. In computer vision, given sufficient exemplars, the characteristics of objects of interest (say people in a crowd) can perhaps be learned.

In the bottom-up framework, filtering comes before segmentation, and can be quite aggressive. Parts we do know are not of interest to us can be removed. Parts we know are wholly inside or outside the object of interest can be kept separately and used as "markers" or "seeds". These will be used as starting point for the contour placement step of segmentation.

(a) (b) (c)

Figure 1.1: A typical computer vision problem: outlining a kangaroo in the bush. In (b) an individual segmentation. In (c) a superposition of several manual segmentations. Contours that are drawn several times appear brighter. Many operators draw the contour of the kangaroo but the level of detail varies widely.

The *top-down* philosophy is a better fit to computer vision than to image analysis, and vice-versa for the *bottom-up* approach. For instance, it is difficult in computer vision to segment a *scene* in the classical sense, i.e. an image with unspecified content, in semantically distinct elements (a person, a background, foliage, etc). Indeed there is generally not enough reference or specification to perform the task reliably. As an illustration, on figure 1.1, we show as scene from the Berkeley Segmentation Database (Martin et al., 2001) containing a kangaroo. This scene was manually segmented by several humans, and the contours they drew were superimposed on top of each other. More "popular" contours appear brighter than those that are drawn a small number of times. We notice that various operators have not segmented the scene in the same way: some have drawn details like the eyes, others just the rough outline of the kangaroo. These differences may come from psychological, cultural or individual differences. This underscores the difficulty of specifying the segmentation problem precisely. In a concise manner, we could say that the problem of segmentation is not specified well enough to be solvable mathematically.

In contrast, many *image analysis* problems are better specified. The goal of an image analysis problem is to get precise measurements, reliable object counts, or indicators of the presence or absence of some feature, from visual data. While the means to attain that goal are not specified, at least the goal itself is. For instance, as illustrated on Fig. 1.2, we have an MRI slice of a human heart, from which we want to detour the left ventricle, which is most often implied in infarcts. From this contours we are able to measure the area of he left ventricle, and if we integrate this measure over several slices of the same heart and in times, we can estimate the volume of blood the heart pumps out every cycle. This particular measurement is useful in medical practice (Najman et al., 2008).

The difference between a computer vision and an image analysis problem is subtle. In particular, both fields of application often use the same tools. It is only in the usage of these tools that things differ. It is worth remembering that in all image analysis problem, the finality is not the segmentation or the recognition, but the final measurements or assessments one wants to make out of image data. The definition of the objects of interests themselves are important, and a more precise definition will lead to better results in applications.

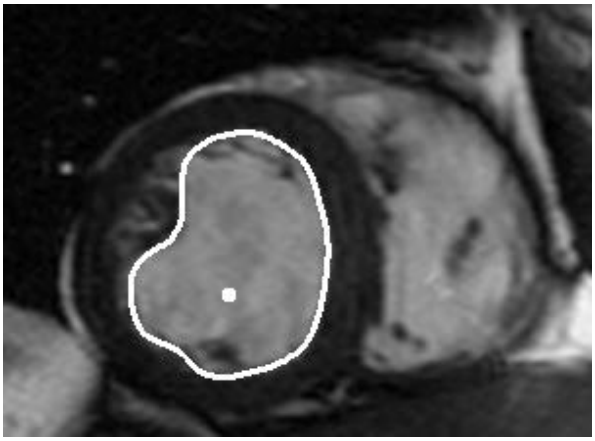


Figure 1.2: A typical image analysis problem: a segmentation of the left ventricle of the heart.

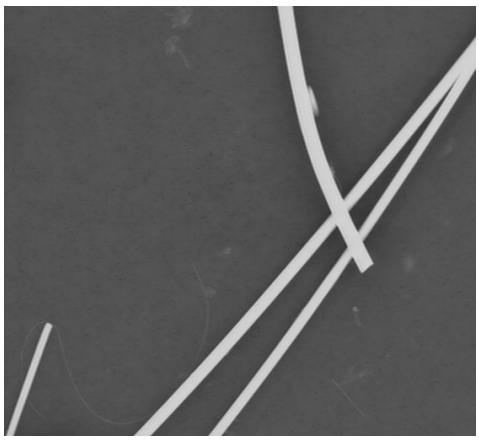


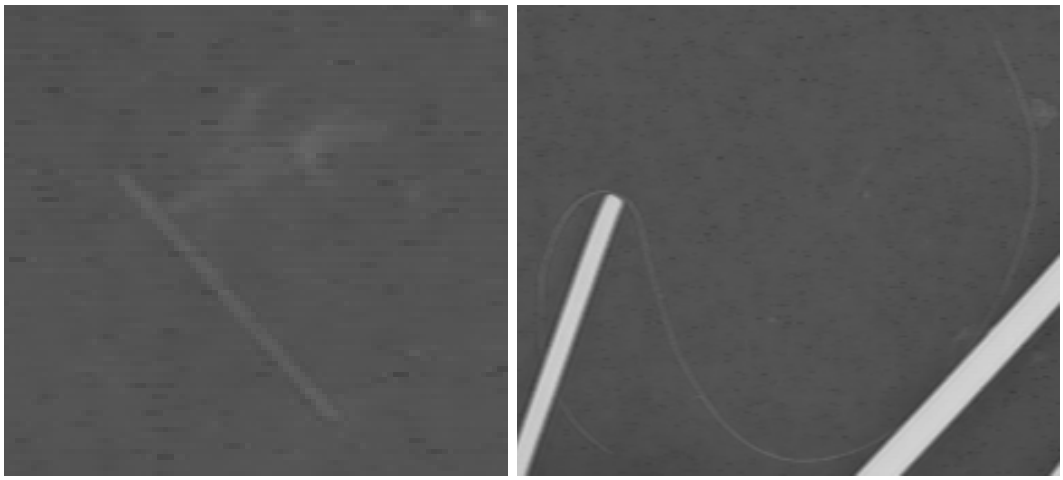
Figure 1.3: An image including several thin objects: some man-made mineral fibre observed in scanning electron microscopy.

1.1.2 Definition of thin objects

In the remainder of this first part, we will restrict ourselves to the problem of analysis of thin objects. Defining a thin object should be simple, for instance we can take as operating definition that a thin object is one that has at least one dimension smaller than the other. In other words, a thin object can be well-approximated by a submanifold of $\mathbb{R}^n$, where n is 2 or 3 typically, of lower dimension than n. For instance a thin fibre in 3D can be approximated by a curve, of dimension 1. We show on a simple example that even though this definition is intuitive and simple, it is not sufficient.

In Fig. 1.3, one can observe several thin white objects on a dark background. One can distinguish four relatively large fibres, of which two have visible extremities, and two not. One can also observe, with more difficulty, two extra very thin fibres, one which is quite short and straight, while the other is longer and curved.

Although they look quite different, the two types of objects (i.e: "large" fibres (they are about 10 μm in diameter, so this is a relative term) and "thin" fibres) do roughly correspond to the definition given above. In reality, this definition is not sufficient to allow for a unified



(a) (b)

Figure 1.4: Details of figure 1.3 : two fibres that are less visible..

Figure 1.5: A thresholding of figure 1.3

segmentation procedure for both kinds of fibres. "Large" fibres are relatively easy to filter and segment: one can readily imagine that a simple thresholding will yield at least the overlap of the large fibres. However, this is not the case for the thin fibres.

In the given example, one is straight and isolated (Fig. 1.4(a)), while the other is curved and partially covered by a large fibre. These fibres are not straightforward to filter or segment. As shown on Fig. 1.5, even careful thresholding does not allow both fibres to be included without noise coming from the background. Additionally, some of this noise looks like thin objects, e.g. horizontal thin lines that are artifacts from the acquisition instrument, and the thin curved fibre is not present in the threshold in its entirety.

The temporary conclusion from this illustrating example is that the definition of a thin object depends on the application. To be able to make progress, we will require a standard allowing the identification of interesting objects with respect to noise. It is unlikely that a universal definition or tool will be available. In addition, we begin to appreciate the concept that thin objects are difficult to deal with when their smallest dimension approach the limit

(a) (b) (c)

Figure 1.6: Scale and local orientation. (a) shows some text, where thin objects are made of the strokes composing letters ; (b) shows the same text blurred by a Gaussian kernel of standard deviation $\sigma = 5$. Here thin objects are the words themselves. (c) shows again the same text, blurred with $\sigma = 10$. Thin objects have all but disappeared.

of resolution of the imaging modality.

1.1.3 Orientation and scale

Another common complicating factor related to resolution is that of scale . Commonly, defining a single orientation at every point of an image is impossible. This is often due to three main effects: (i) in 2D it is possible that several objects cross each other[1], (ii) in any dimension, a branching may occur (as in a vessel) and (iii) local orientation may depend on the scale objects are considered. This latter effect is illustrated on Fig. 1.6.

Without involving any blur, it is clear that scale and orientation are linked. Consider for instance a blood vessel imaged in MRI. For a larger artery like one of the carotids, at a large scale it is possible to discern the outer layer of the vessel, which locally is a surface-like thin object. However at a smaller scale, the vessel itself is a line-like thin object, as illustrated on Fig. 1.7. Deciding the nature and orientation of thin objects is therefore a difficult problem in itself.

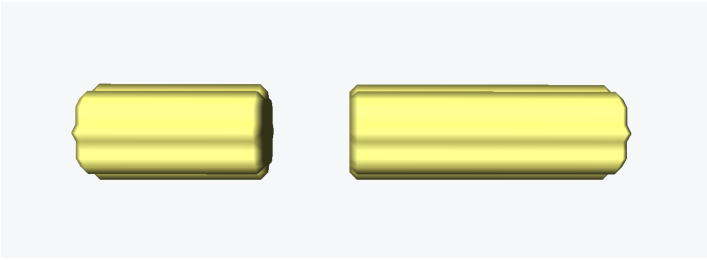
1.2 Application fields

In spite of the lack of a usable generic definition, it is probably useful to illustrate the field of application of the operators we will later describe through the use of examples. These cannot hope to be exhaustive, but at least can be representative of a sufficiently large class of problems.

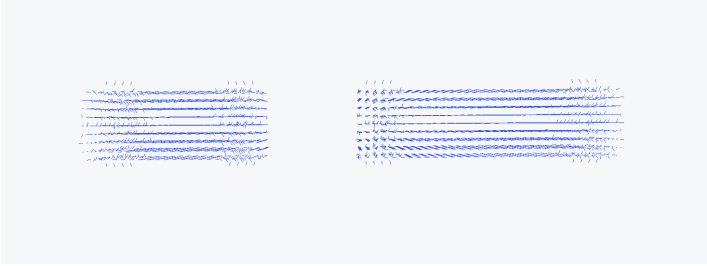
1.2.1 Some 2D problems

In the example above, the problem was to segment all the fibres in order to measure their length and diameter. We show some other related examples, in order to illustrate some of the various difficulties encountered in dealing with thin objects.

[1]In 2D this is a common occlusion problem due to projection, in 3D it can only happen if the material under study allow interpenetration, which is possible but not common.

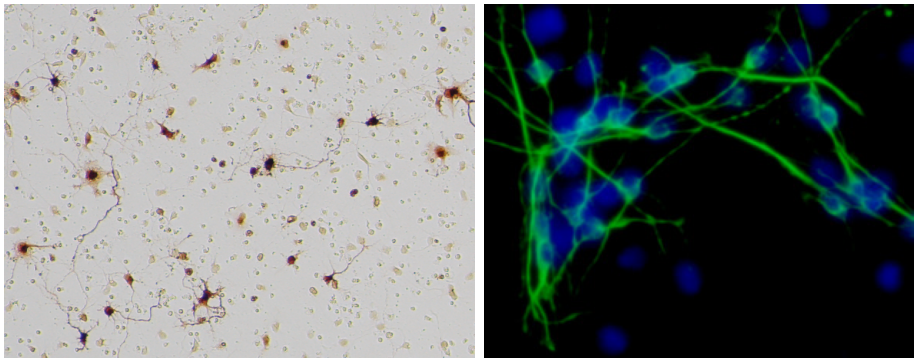


(a)



(b)

Figure 1.7: Discrete cylinders and their scale-dependent local orientation. Orientations near the boundary of the cylinders are normal to the surface, while the orientations near the axis of symmetry of the cylinders are oriented along the axis.



(a) (b)

Figure 1.8: Images of neurites. (a) Note the thinness and tortuosity of the dendrites.(b) image of overlapping neurites in fluorescence microscopy

1.2.1.1 Neurites

Neurites are a generic term to describe projections from cell bodies of neurons, i.e. axons and dendrites. Dendrites and axons in 2D microscopy tend to be quite tortuous (see Fig. 1.8(a). The assumption of local smoothness is not verified in this example.

The individual neurites can be generally assumed each to possess a tree structure, although this is not strict. In addition, neurites seldom appear in isolation, leading to overlapping structures, as in Fig. 1.8(b). In fluorescence modalities, disconnections are also common.

In spite of this, we might still hope to filter and segment the neurite structure under study,

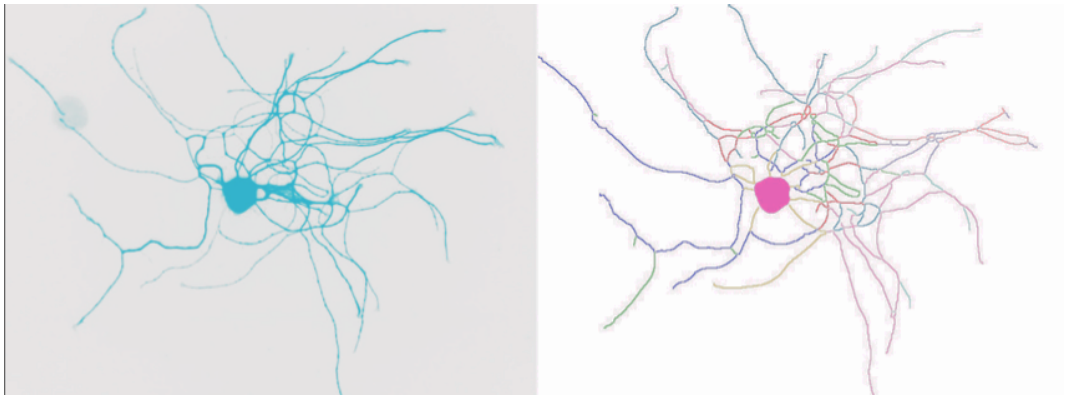


Figure 1.9: Segmentation and identification of neurites in HCA-image.

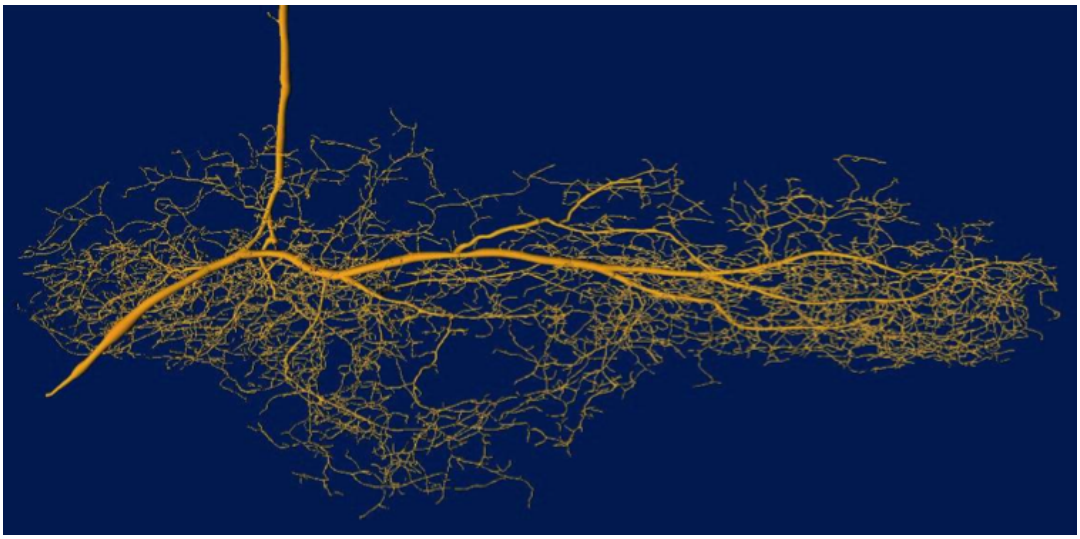


Figure 1.10: A 3D view of an in-vivo *moth neurite.*

as on Fig. 1.9 using the package HCA-image, which we helped develop at CSIRO, Australia.

Of course, neurites *in-vivo* are really 3D structures (see Fig. 1.10). The structure of the dendrites tree becomes quite complex. This illustrates the fact that the real structure of the neurites is resolution-dependent. As the imaging resolution increases, so does the complexity of the network. We will see that this is often the case in natural structures. This remark implies that we cannot simply solve the problem of natural thin structures by increasing the resolution until the objects under study are no longer thin. Not only will we inevitably bump into natural imaging limits (for instance because of the frequency of light), but also will we also uncover another layer of complexity. Instead of making our life easier, increasing the resolution makes it harder.

1.2.1.2 Retina blood vessels

Images of the retina are useful for the diagnosis of many diseases. These include of course eye-related conditions, such as macular degeneration, but also others that have circulatory implications such as diabetes. One reason for this is that the eye fundus shows the capilatory

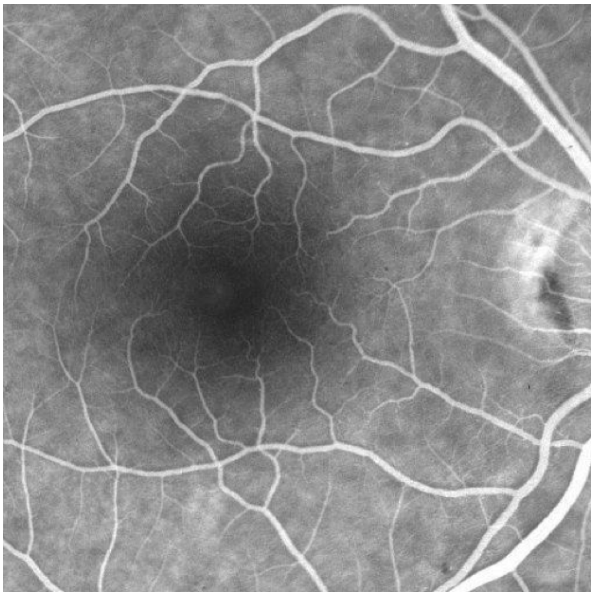


Figure 1.11: A image of the eye fundus showing the blood vessels.

network very well.

Such images exhibit a very thin tree-like vascular structure over an uneven background. The eye fundus follows the general shape of the eye, which is roughly spherical. The fundus is relatively thin but has some 3D features.

Segmeting this network of blood vessels is an important problem, particularly when looking for micro-aneurisms, which are blockages in this network. They are a marker for diabetes and various cardio-vascular diseases. Similarly to the previous neurite problem, the complexity of the blood vessel network in the eye fundus increases with the quality of the acquisition. We are faced with the task of delineating an object that shows increasing detail all the way to the resolution limit of the acquisition.

1.2.1.3 X-ray images of stent guide

A stent is a cylindrical, hollow, meshed object that is deployed in a blood vessel to enlarge its lumen. The use of stents, while delicate, is much more benign than conventional surgery. It has revolutionized the treatment of artery blockage particularly around the heart. Stent guides are wire-like objects introduced in the larger blood vessels, that are used to push an undeployed stent to the place where it is needed via the arterial network.

In these kinds of images, the noise levels can be quite high, because X-ray doses must be minimized during this interventional procedure and so photon count is low. As shown on Fig. 1.12, the guide is hardly visible throughout its entire length. As well, many simarly looking objects are visible in the image, such as rib edges. These images are indeed particularly challenging.

1.2.1.4 Oriented textures

While images shown before come from specialized domains, thin objects are also part of everyday life. In so-called natural images, they often appear as part of textures. It has been

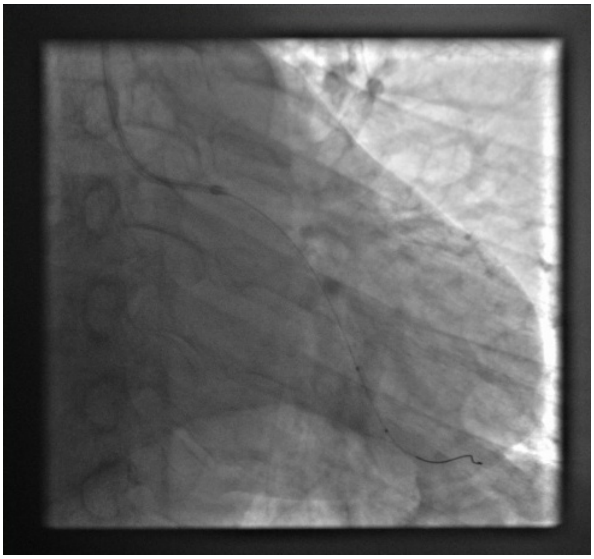


Figure 1.12: An image of a stent guide in X-ray fluorescence imaging.

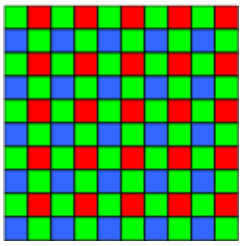
Figure 1.13: Classical test colour image of Barbara, with oriented textures.

long recognized that textures require specific procedures for denoising. The classical image of "Barbara" (see Fig. 1.13) with her striped outfit, but also the rattan chair behind her is often used in advanced denoising examples.

This image is not particularly noisy, but when used as a filtering benchmark, example noise is simulated. The objective is to recover the image without removing the texture information.

1.2.1.5 Oriented elements in photography

A problem related to texture is the digital demosaicing problem. In most digital cameras, each pixel records only one colour channel out of three. Consequently interpolation must be used



(a)

(b) (c)

Figure 1.14: Most modern digital cameras use a colour filter array (CFA) like the one in (a), using a single sensor but causing only one colour channel to be captured at each pixel. Interpolation of nearby pixels must be performed to recover the other two channels. However using simple interpolation schemes causes artefacts on thin objects as in (b), (c).

to recover the full colour image. However simple isotropic interpolation tends to yield strong colour artifacts when the resolution of features on the image comes close to the sampling frequency. In this case it is very useful to interpolate pixel data in a directional way.

On Fig. 1.14, we show one example of colour filter array (), in this instance the Bayer CFA, which is the most common type. The lighthouse image is part of the Kodak 1990's era PhotoCD demo collection and is often used to showcase interpolation artefacts because the vertical elements of the fence in the picture. To achieve better demosaicing results, one must make use of the local orientation, and to perform the colour reconstruction along the local orientation instead of across it.

1.2.2 2D+t problems

Thin objects can also be defined in time. A compact, moving object can be viewed as a thin object problem in 2D+T.

(a) (b) (c) (d) (e)

Figure 1.15: Some frames of the "table tennis" CIF test video sequence, originally meant for video coding.

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1.16: Some frames from the Zhang Yimou feature film "House of Flying Daggers" (`http://www.imdb.com/title/tt0385004/`). Some scratches and artifacts are obvious and appear in a single frame (see (b)). Others, like on the forehead of the heroine, only appear when viewing consecutive frames.

1.2.3 Tracking small compact objects in time

In Fig 1.15, the trajectory of the ping-pong ball forms a thin object in 2D+t, which is a useful representation for tracking. However, one difficulty is the fact that most often, the temporal resolution of the film is much less than the spatial, and so object discontinuities are likely.

1.2.4 Injected X-ray angiography

A different object is one which is already thin in 2D and moving in time, as in in fact the case for Fig. 1.12. Indeed, in this problem the stent is moving in time with the beating of the heart. In this case the object of interest is thin in 2D+t, it creates a surface.

A similar example is that of a scratch on the surface of a damaged film reel. An example is shown in Fig. 1.16, which shows some frames of the unique edited master copy on film, which was damaged in the film digitizer . While some examples of artefacts are essentially 2D and are easy to spot. Others only appear when taking into account the time domain. They are typical thin objects in the 2D+t domain, which means they cannot be easily corrected only using frame by frame processing.

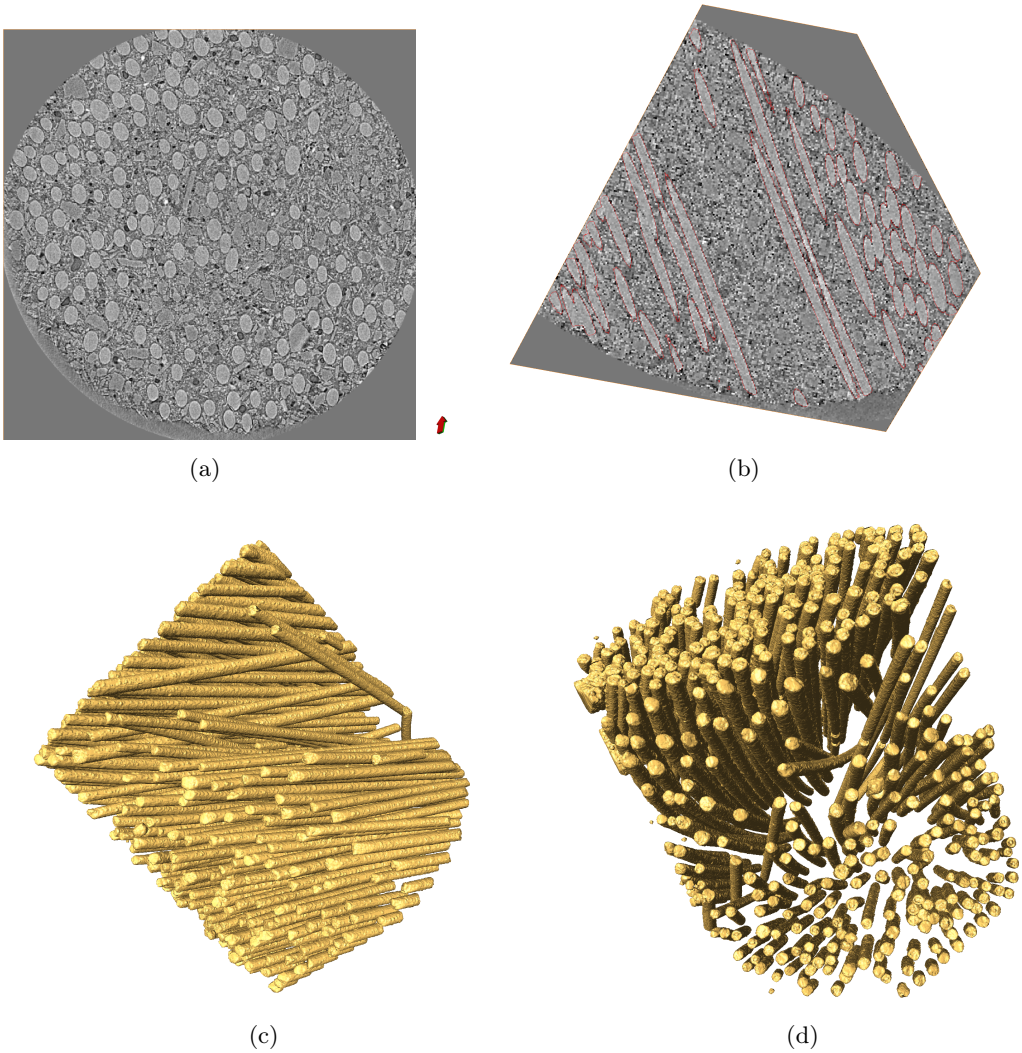


Figure 1.17: Reinforcement fibres in a substrate. Slices through the material (a) and (b) and associated segmentation (c) and (d).

1.2.5 3D problems

There exists a still greater variety of thin objects in 3D. Essentially thin objects can be of codimension one or two. We start with codimension two examples, i.e. objects that are thin in two dimensions.

1.2.5.1 Fibres in 3D

Fibres and fibrous objects are common in materials science, for instance reinforcement fibres in compound materials.

In Fig. 1.17, we show a high resolution X-ray micro-CT view of a fibre-reinforced compound material. Even though this image has enough resolution to show the inside of the fibres to some extent, it is still useful to view it as a thin object. Because these fibres are significantly longer

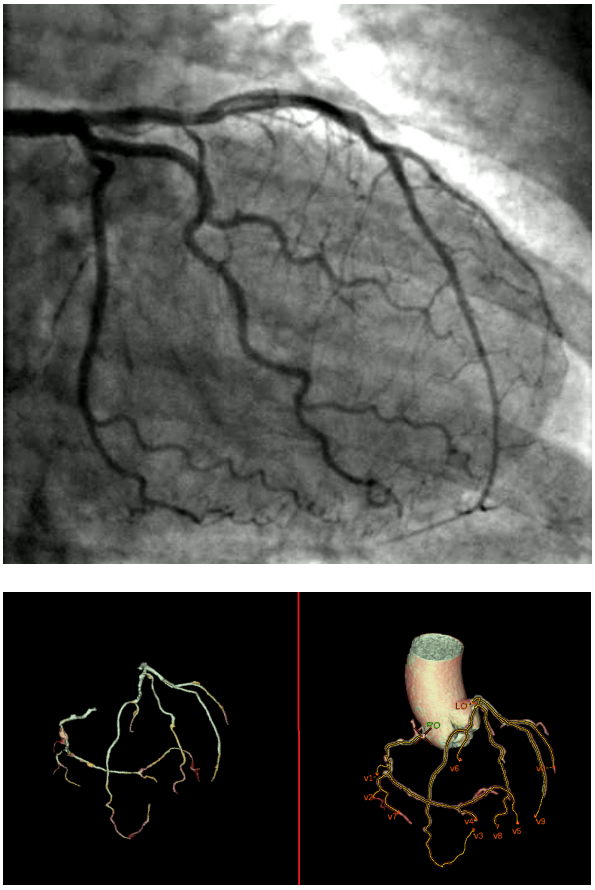


Figure 1.18: A projection view of the heart coronaries in CT injected numerical angiogram in (a). In (b), a current state-of-the-art segmentation of the main arteries.

than they are wide, many segmentation methods based on perimeter/surface minimization may fail on these objects.

1.2.5.2 Vessels in medical imaging

A more difficult problem similar to that in section 1.2.1.2, but in 3D this time is the challenge of 3D angiography, i.e. the filtering and segmentation of vessels in various organs under various imaging modalities. Heart coronaries angiography (coronography) is useful in the lead up to interventions to prevent or heal infarction, or during screening for this condition.

In Fig. 1.18, we show a 2D projection view of the coronaries, and simultaneously a current, state-of-the-art segmentation of the main coronary arteries. We can see that the discrepancy in the level of detail is substantial.

Brain vessel imaging is also common in many lead up to brain surgery, e.g. in the case of aneurisms, brain tumours or arterio-venous malformations. In the case of the brain, even small aneurysms may have serious consequences, however, as shown on Fig. 1.19, the brain blood vessel system is particularly complex.

We face the same problem as before in 2D. Vessel system are very complex and will challenge any system all the way to the highest resolution that they support. Consequently,

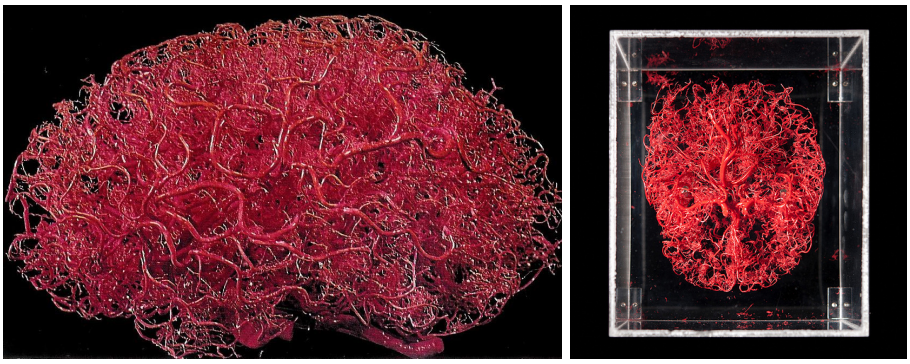


Figure 1.19: A so-called "corrosion cast" of the blood vessels in the brain. The level of details required to image this structure in 3D in vivo *is beyond the capabilities of current imaging devices.*

filtering and segmentation task will continue to difficult and relevant.

Many other biological structures are made of thin objects, such as the airway structure in the lungs.

1.2.5.3 Cracks in concrete

Codimension 1 problems in 3D also clearly exist. They correspond to problems concerning surface-like objects, i.e. that are thin in one dimension only.

An example is given on Fig. 1.20, which shows a network of cracks developing in a test slab of concrete. Cracking is obtained by fast drying. Such cracks are sometimes smooth as they follow the contours of inclusions (here glass spheres) and sometimes very irregular, showing tunnels and disconnections. Segmentation of such structures is particularly challenging as a result.

1.2.5.4 Seismic data, fracture lines

Many other codimension 1 thin structures exist in nature, for instance the layered structures of seismic data, as shown on Fig. 1.21.

1.2.6 3D+t problems

3D+t problems are often at the frontier of what is feasible today in terms of sheer data size. This is sometimes compounded in the case of thin objects detection, since a high resolution is most often needed to ensure good results.

1.2.6.1 Cracks in concrete over time

To illustrate this issue, in fact the crack problem of section 1.2.5.3 is really a 3D+t problem. Samples were taken at drying times $t = 0$, $t = 24$h and$t = 48$h, and current procedures for crack detection (Bernard et al., 2012) do use the 3 different times. However these images samples are quite large. Each independent image, or frame, has a full resolution size of

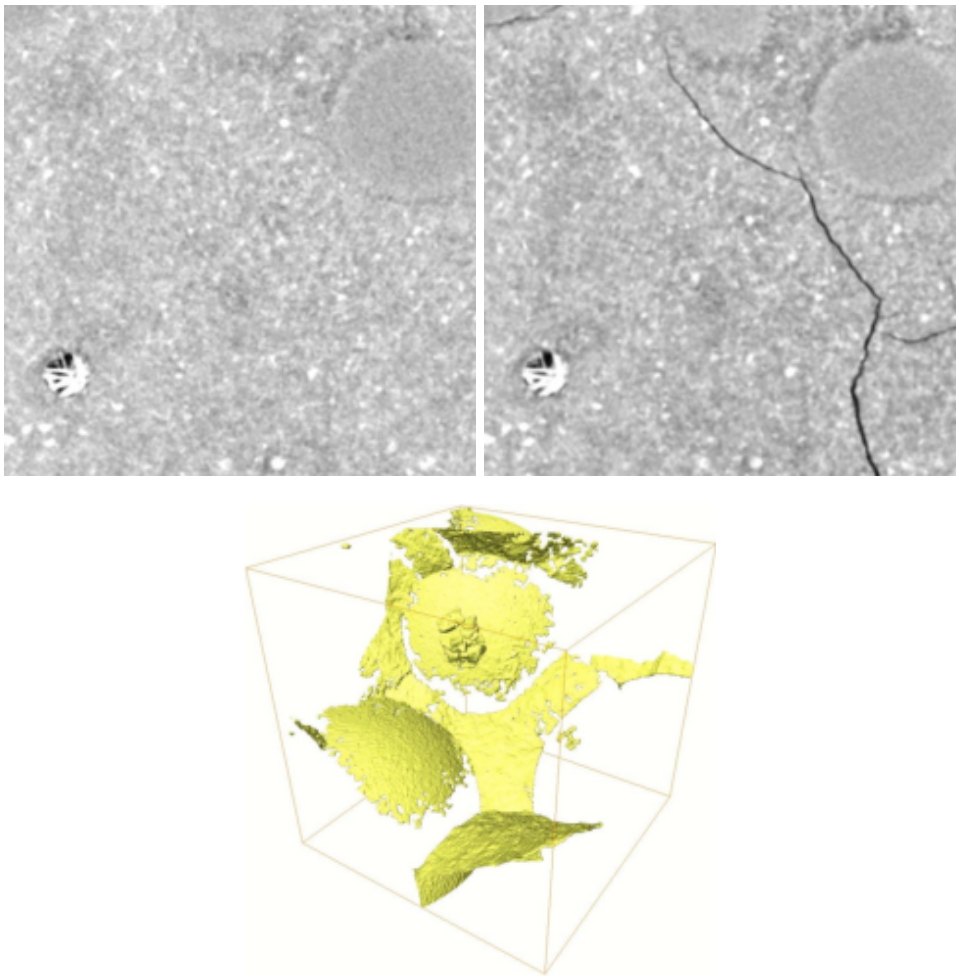


Figure 1.20: Cracks developing in concrete through fast drying: (a) Initial image at t=0 ; (b) cracks at t=24h ; (c) 3D rendering of the crack network. Data courtesy CNRS-ICMCB.

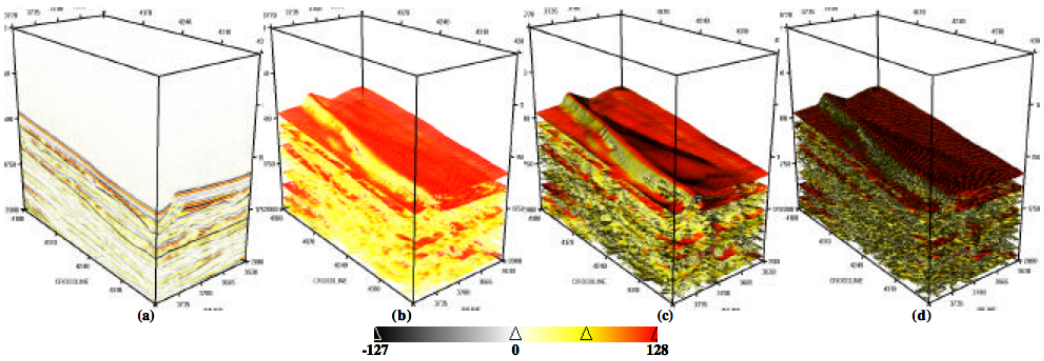


Figure 1.21: Seismic data rendering, courtesy Petrobras

Table 1.1: Some characteristics of thin objects

Example	Problem	Difficulties
2D fibres	Measuring lengths and diameters	Variability in dimensions, very thin objects
2D Neurites	Length and branching numbers	Non locally straight, overlapping structures, disconnections. True structure is resolution-dependent, objects remain thin whatever the resolution..
Retina blood vessels	Segmentation	Uneven background, high branching number, high resolution problem.
Stent guide	Noise reduction, tracking	High noise, moving object, uneven background, similar objects in the background
Oriented textures	Filtering, identification	Wide variability of appearance and noise levels.
CFA demosaicing	Reconstruction, filtering	Resolution limited, sensitivity to noise.
Tracking small objects	Coding, tracking, stereo	Time-domain disconnections
X-ray fluorescence angiography	Denoising, tracking	High noise levels
Film scratches	Inpainting	Time-domain disconnections
3D fibres	Filtering, segmentation	Large relative surface breaks most segmentation models
3D vessels	Filtering, segmentation	Dense thin structure, very high relative surface, complexity
Concrete cracks	Detection	Very high resolution needed, large data size, noise levels.
Seismic data	Detection, segmentation	Disconnections, scene complexity
3D cracks over time	Detection, segmentation	Noise levels, resolution, even higher data size

$2048^3 \times$ 16 bits, i.e each frame is a 16GB dataset. Processing the full resolution image is required as some of the cracks have sub-voxel size even at that resolution.

1.3 Summary

We summarized the examples we studied on table 1.1.

Based on this table, thin objects are prone to extra problems when compared with more "compact objects":

- Increased sensitivity to noise levels;
- disconnections more likely;
- wide variability of configurations;

- resolution and scale-related problems;
- traditional filtering and segmentation models do not fit.

It should be relatively clear now that thin objects require specific filtering and segmentation procedures. In the next chapter we will briefly survey existing methods and introduce some of our contributions.

Chapter 2

Thin object filtering and segmentation

In this chapter we survey classical methods for thin object analysis and we propose some contributions. Evidently the need for segmentation, noise reduction in the presence of thin objects, CFA demosaicing and so on has a long history. In very rough terms, most approaches have essentially consisted in estimating the local orientation of the image content, and to design operators that use this information in some way. From some the examples in the previous chapter, it should be also clear that "local" orientation is something that is scale-dependent.

2.1 Linear filtering

Linear filters that take into account orientation or texture fall in a small number of categories: (i) those that explicitly first estimate a local orientation field as a tensor, potentially in a multi-resolution approach; (ii) those that first decompose images in a multi-resolution fashion, possibly including orientation; and (iii) filters that use non-local information.

2.1.1 Local derivative-based oriented filters

There are two widely used, derivative-based filters that can be used to estimate orientation: the *Hessian* and the so-called *structure tensor*. Both are derived from the standard gradient operator. It is useful to study the gradient operator first to see why more complicated tensor field are in fact useful in the case of thin objects.

2.1.1.1 The gradient operator

The gradient operator is the standard 1-form (or covector):

$$\nabla(I) = \begin{bmatrix} \frac{\partial I}{\partial x_1} \\ \vdots \\ \frac{\partial I}{\partial x_n} \end{bmatrix} \tag{2.1}$$

The gradient defines a vector field and so can be associated with an orientation. Since the gradient is a linear operator, it is susceptible to linear scale-space theory and can thus be used in a multi-scale fashion. However, the gradient operator can only be used to determine the orientation of the edges of features, not of the features themselves. In addition, using a

2D		3D			Local profile
λ_1	λ_2	λ_1	λ_2	λ_3	
Large+	Large+	Large+	Large+	large+	dark blob
Large-	Large-	Large-	Large-	Large-	light blog
Small	Large+	Small	Large+	Large+	dark string
Small	Big-	Small	Large-	Large-	light string
		Small	Small	Large+	dark plane
		Small	Small	Large-	light plane

Table 2.1: Possible eigenvalue responses and their signs after the eigenanalysis corresponding to different shapes and color intensities.

multiscale gradient tend to average out the orientation information, with less-than-desirable results. This is illustrated later in the text in Fig. 2.2. These properties make it desirable to define other operators, in order to achieve more useful orientation scale-spaces.

2.1.1.2 The Hessian operator

The *Hessian* operator is simply the matrix of all second derivatives. For a n-dimensional image $I(x_1, x_2, \ldots, x_n)$, assuming continuity, the Hessian $\mathcal{H}$ is a square, symmetric matrix given by:

$$\mathcal{H}(I) = \begin{bmatrix} \frac{\partial^2 I}{\partial x_1^2} & \frac{\partial^2 I}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 I}{\partial x_1 \partial x_n} \\ \frac{\partial^2 I}{\partial x_2 \partial x_1} & \frac{\partial^2 I}{\partial x_2^2} & \cdots & \frac{\partial^2 I}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 I}{\partial x_n \partial x_1} & \frac{\partial^2 I}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 I}{\partial x_n^2} \end{bmatrix} \tag{2.2}$$

The interpretation of the Hessian is relatively simple. If at some point x the derivatives of I are all zero, and if the Hessian is positive definite at x, then $I(x)$ is a local minimum of I. Conversely, if it is negative definite at x, then $I(x)$ is a local maximum. If at x, H has both positive and negative eigenvalues, then x is a saddle point. At points where the derivatives of I do not vanish, $H(I)$ is still useful, mostly by studying its eigenvalues.

On Fig. 2.1, we show a figure from Danielsson and Lin (2001), where g is the "derotated" version of I, i.e. I in a referential where the principal eigenvectors form the basis. It is possible, by using comparisons between eigenvalues, to distinguish locally between various elementary shapes. Of particular interest to us are what Danielsson and Yin call the "string" and plane detectors.

We denote the three eigenvalues of $\mathcal{H}$ $\lambda_1, \lambda_2, \lambda_3$, ordered such that $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$, as well as $\mathbf{e_1}, \mathbf{e_1}, \mathbf{e_3}$ their corresponding eigenvectors. the eigenvalues represent the principal curvatures and the eigenvectors their direction. We see that if at some point one eigenvalue is small whereas the other two are large with the same sign, then locally the image looks like a ridge, i.e. a string or line-like feature.

In general, eigenvalues and their signs can be used to distinguish features, following the qualitative classification shown in Table 2.1.

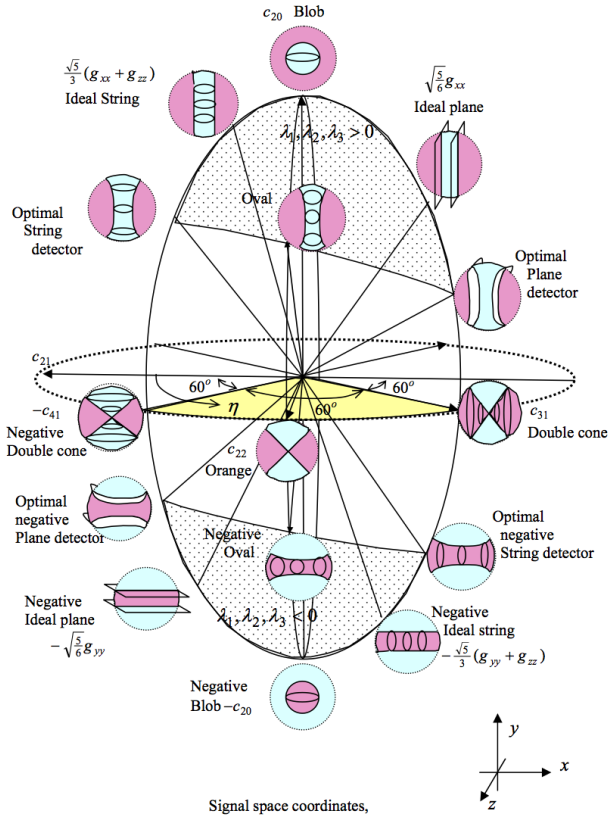


Figure 2.1: Eigenvalue-based analysis of the 3D Hessian, see Danielsson and Lin (2001) for details.

2.1.1.3 Multi-resolution - Scale-space - Vesselness functions

One problem with using derivative-based information is to cope with noise, another is to combine information at different scales. As shown in chapter 1, section 1.1.3, orientation and shape classification is scale-dependent; and it is well-known that noise becomes an increasingly important issue with the order of derivation. It is possible to offer a solution to both problems by considering a Gaussian multi-resolution pyramid, following linear scale-space theory (Witkin, 1983; Lindeberg, 1994). However, it is not always easy to combine information from different scales into one measure.

One approach, for example for finding string-like objects, possibly with junctions and bifurcations, as in vascular networks, is to use a *vesselness* function. Several have been proposed, but the most common are the Frangi and the Sato vesselness.

The Frangi vesselness For a 3D image $I(x)$, $x \in \mathbb{R}^3$ observed at a scale σ, The Frangi et al vesselness function (1998) is given as follows:

$$\nu(x,\sigma) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0, \\ (1 - e^{\frac{-R_A^2}{2\alpha^2}}) \cdot e^{-\frac{R_B^2}{2\beta^2}} \cdot (1 - e^{\frac{-S^2}{2c^2}}) & \text{otherwise,} \end{cases} \tag{2.3}$$

with

$$\begin{aligned} R_A &= \frac{|\lambda_2|}{|\lambda_3|}, \\ R_B &= \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}}, \\ S &= \|H_\sigma\| = \sqrt{\textstyle\sum_j \lambda_j^2}, \end{aligned} \tag{2.4}$$

in which R_A differentiates between plane- and line-like objects, R_B differentiates vs. blob-like ones, and S accounts for the intensity difference between objects and background. Parameters α, β and c influence the sensitivity of the filter to the corresponding measures.

For 2D images, the Frangi et al. vesselness is of course simpler, and can be expressed as:

$$\nu(x,\sigma) = \begin{cases} 0 \quad \text{if } \lambda_2 > 0, \\ e^{-\frac{R_B^2}{2\beta^2}} \cdot (1 - e^{\frac{-S^2}{2c^2}}) \quad \text{otherwise,} \end{cases} \tag{2.5}$$

and

$$\begin{aligned} R_B &= \frac{|\lambda_1|}{|\lambda_2|}, \\ S &= \|H_\sigma\| = \sqrt{\textstyle\sum_j \lambda_j^2}, \end{aligned} \tag{2.6}$$

The Sato vesselness Similarly, Sato et al. in 1998 proposed the following measure:

$$S(x) = \begin{cases} \sigma^2 \left|\lambda_3\right| \left(\frac{\lambda_2}{\lambda_3}\right)^\xi \left(1 + \frac{\lambda_1}{|\lambda_2|}\right)^\tau, \lambda_3 \leq \lambda_2 \leq \lambda_1 < 0 \\ \sigma^2 \left|\lambda_3\right| \left(\frac{\lambda_2}{\lambda_3}\right)^\xi \left(1 - \rho\frac{\lambda_2}{|\lambda_2|}\right)^\tau, \lambda_3 \leq \lambda_2 \leq 0 \leq \lambda_1 \leq \frac{\|\lambda_2\|}{\rho} \end{cases} \tag{2.7}$$

where $\xi \geq 0$ influences cross-section asymmetry, $\tau \geq 0$ controls the sensitivity to blob-like structures, $0 < \rho \leq 1$ controls sensitivity to the tubular object curvature, and σ^2 normalizes responses across scales.

Both the Frangi and the Sato vesselness yield a number between 0 and 1, which can be viewed as a probability of a pixel to belong to a "vessel", i.e. a tubular object, possibly with bifurcations. The vesselness function can be computed at various scales (i.e. various values of σ), and the strongest response selected.

Several other vesselness functions, based on the eigen-analysis of the Hessian, have been proposed. One may cite the work of Manniesing et al. (2006), who developed a continuous version of Frangi's vesselness, suitable for oriented diffusion.

2.1.1.4 Structure tensor

The *structure tensor* $\mathcal{J}$ is the matrix of second-order moments of the gradient of an image I. It is defined as follows:

$$\mathcal{J}(I) = \begin{bmatrix} \left(\frac{\partial G_\sigma(I)}{\partial x_1}\right)^2 & \frac{\partial G_\sigma(I)}{\partial x_1}\frac{\partial G_\sigma(I)}{\partial x_2} & \cdots & \frac{\partial G_\sigma(I)}{\partial x_1}\frac{\partial G_\sigma(I)}{\partial x_n} \\ \frac{\partial G_\sigma(I)}{\partial x_2}\frac{\partial G_\sigma(I)}{\partial x_1} & \left(\frac{\partial G_\sigma(I)}{\partial x_2}\right)^2 & \cdots & \frac{\partial G_\sigma(I)}{\partial x_2}\frac{\partial G_\sigma(I)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial G_\sigma(I)}{\partial x_n}\frac{\partial G_\sigma(I)}{\partial x_1} & \frac{\partial G_\sigma(I)}{\partial x_n}\frac{\partial G_\sigma(I)}{\partial x_2} & \cdots & \left(\frac{\partial G_\sigma(I)}{\partial x_n}\right)^2 \end{bmatrix}, \tag{2.8}$$

where $G_\sigma(I) = G(\sigma) \star I$, with $G(\sigma)$ the centered Gaussian with variance σ^2 and $\star$ the convolution operator. Alternatively, the structure tensor can be written as an outer product,

$$\mathcal{J}(I) = \nabla(G_\sigma(I)).\nabla(G_\sigma(I))^\intercal, \tag{2.9}$$

with ∇ the standard gradient operator written as a $n \times 1$ (single-column) covector.

The structure tensor (ST) summarizes the predominant directions of the gradient in a neighborhood. As such, it can be used in much the same way as the Hessian, although its interpretation is different. Since the determinant of the ST is a quadratic form, and due to its structure, the ST is symmetric positive semi-definite. All the eigenvalues of the ST are non-negative, and can be ordered as before, e.g. in the 3D case $0 \leq \lambda_1 \leq \lambda_2 \lambda_3$.

In this instance, λ_3 is the largest eigenvalue and $\mathbf{e_3}$ provides the principal direction of the gradient of I. If $\lambda_3 >> \lambda_1 \approx \lambda_2$, then this means that locally the iso-level of I are surface-like perpendicular to $\mathbf{e_3}$. If $\lambda_3 \approx \lambda_2 >> \lambda_1$, then the isosurfaces tend to be tube-like parallel to $\mathbf{e_1}$. If all three eigenvalues are similar, this means that locally the gradient isosurfaces are sphere (blob)-like. By themselves, the eigenvalues do not allow to distinguish whether the structures are bright or dark. Similar to the Hessian case, multiscale can be handled through varying σ.

2.1.2 Filter banks

In the previous section, we introduced operators that allowed users to measure orientation and provide filters that were able to take into account this information. However orientation is not explicitly measured or taken into account in these operators. We were able to exploit it as a by-product via eigen-analysis.

There exist in the literature operators designed for oriented filtering. There are essentially two categories of operators: those that extend the separability property of the gradient-based operators to achieve a continuous estimation of the orientations, and those that discretize and sample orientation.

2.1.2.1 Steerable filters

Steerable filters (Freeman and Adelson, 1991) are convolution kernels that are designed to be orientation-selective. They can be expressed via a linear combination of a small set of rotated versions of themselves. The name comes from "beam steering", which is use in radio technology such as RADAR. A simple example is the gradient, which yields the structure tensor as above, or the second derivatives, which yield the Hessian. However extension of this idea can yield filters similar to the Gabor filters, which we will introduce shortly.

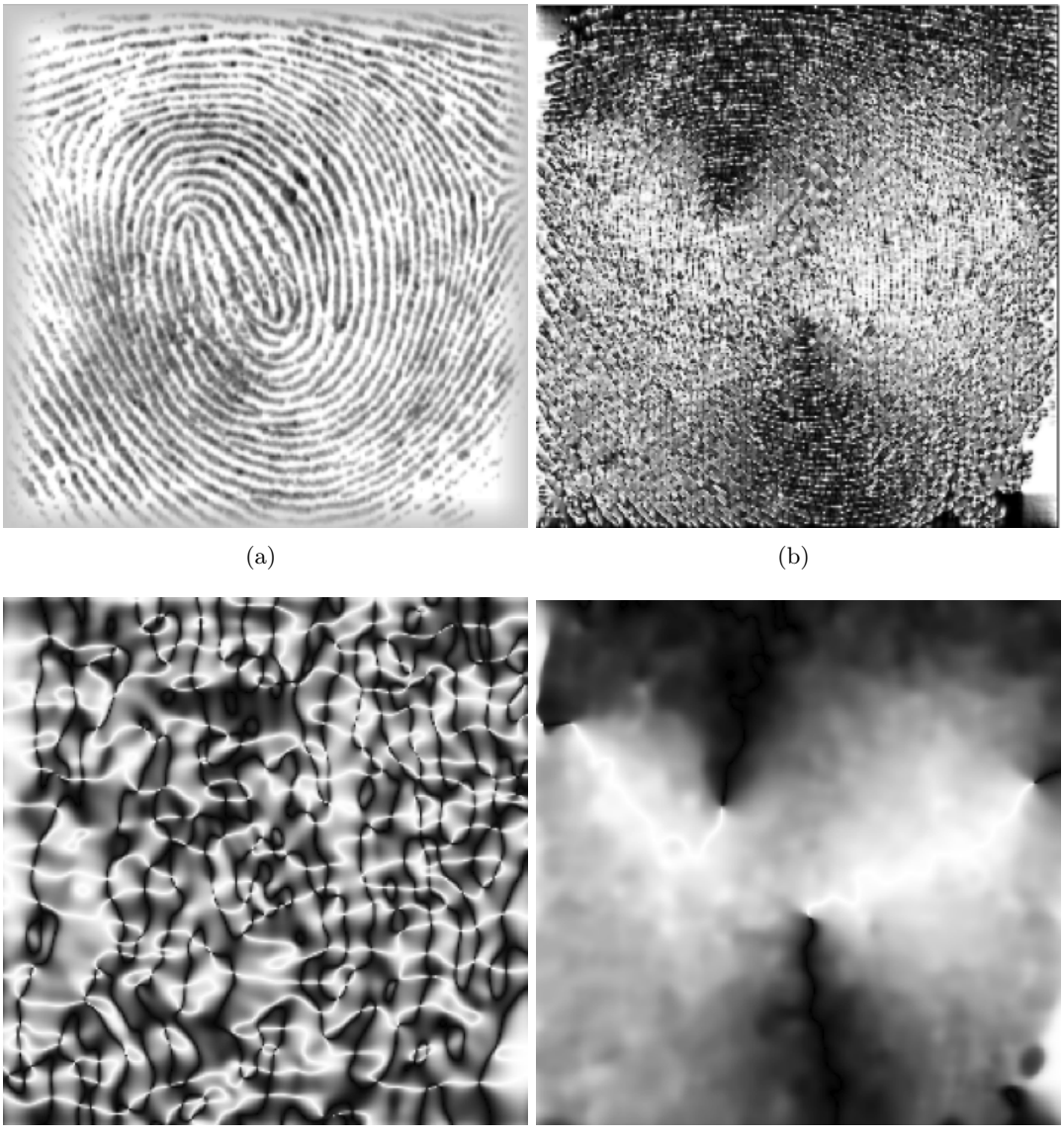
To see that the derivative is steerable, following the example of Freeman and Adelson, let $G(x, y) \equiv \exp -(x^2 + y^2)$ be the 2D Gaussian. The first derivatives are as follows:

$$G_1^0 = \frac{\partial G}{\partial x}(x, y) = -2xG(x, y) \tag{2.10}$$

$$G_1^{\frac{\pi}{2}} = \frac{\partial G}{\partial y}(x, y) = -2yG(x, y) \tag{2.11}$$

Then for any θ,

$$G_1^\theta = \cos\theta G_1^0 + \sin\theta G_1^{\frac{\pi}{2}} \tag{2.12}$$



(a) (b)

(c) (d)

Figure 2.2: An image of fingerprints (a). In (b), local orientation as measured by a slightly smoothed gradient with $\sigma = 1$: orientation information is present but is noisy. In (c), local orientation as measured by a strongly smoothed gradient with $\sigma = 5$: orientation information is all but deleted. In (d), orientation measured by the structure tensor with $\sigma = 1$: it is much more usable. Note that the orientation of both dark and white features are not differentiated. Image courtesy J. Weickert (1998).

Any linear filter based on G_1^θ, e.g. a convolution, can be computed for any θ by projecting onto the basis $\{G_1^0, G_1^{\frac{\pi}{2}}\}$. Of course not every convolution kernel is steerable in this way, but some interesting ones are, for instance wedge filters (Simoncelli and Farid, 1996), ridge filters (Jacob and Unser, 2004), scale-space pyramids (Karasaridis and Simoncelli, 1996), orientation-scale filter banks (Perona, 1995), as well as some wavelet bases (Koren et al., 1995). The steerable property should be quite desirable for 3D filtering, however progress in this area has been slow (Derpanis and Gryn, 2005), because designing such filters beyond the obvious ones (based on derivatives) is harder than in 2D (Unser and Van De Ville, 2009). Some 3D steerable filters do exhibit better performances than the simple ones however (González et al., 2009). In general the steerable property is quite desirable for computing performance, since only the computation of the projections onto the basis is required, but often leads to reduced *filtering* performance compared with filter banks, which we study next.

2.1.2.2 Directional filter banks

Filter banks are a very general concept where the output from various filters are combined in some way. Usually linear filters are involved (convolutions, FIR, IIR, etc), and they are combined in a pyramid or additively. Maximum, minimum or median response is often used as well, as is the combination in a feature vector, typically for texture identification / segmentation.

One very flexible class of oriented, convolution-based filters are the Gabor wavelets (1965). In 1D, a Gabor wavelet $\varphi(t)$ is a Gaussian modulated by a sinusoidal:

$$\varphi(t) = \exp(-\sigma^2 t^2) \exp(j2\pi f_0 t) \tag{2.13}$$

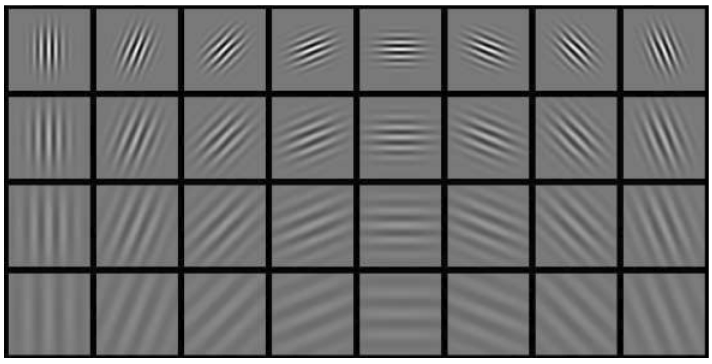
$$\Phi(f) = \sqrt{\frac{\pi}{\sigma^2}} \exp(-\frac{\pi^2}{\sigma^2}(f - f_0)^2), \tag{2.14}$$

where σ is a scale factor, f_0 the modulated centered frequency of $\varphi(t)$, and $\Phi(f)$ is its Fourier transform. The Fourier version is very simple, since it is simply a non-centered Gaussian. In 2D, an individual wavelet can be expressed as:

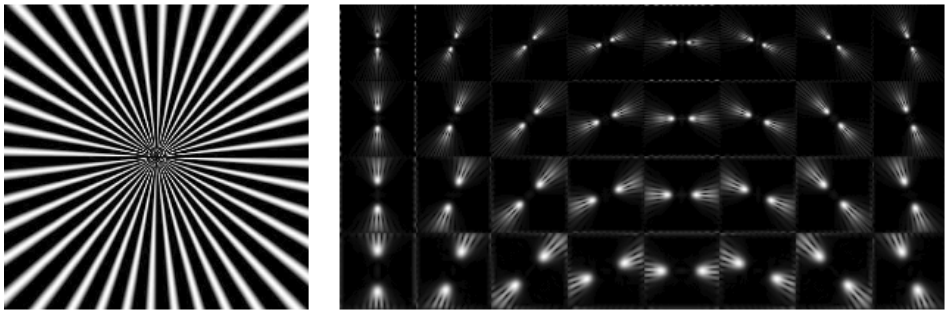
$$\varphi(\vec{z}) = \frac{1}{2\pi} \frac{\|\vec{z}\|^2 \|\vec{k}\|^2}{2\sigma^2} \exp \vec{z}\vec{k}, \tag{2.15}$$

where $\vec{k} = 2\pi f \exp(j\theta)$. This wavelet depends on 5 parameters, which can be expressed as scale, offset (2 parameters), orientation and aspect ratio. Practitioners typically endeavour to cover the whole frequency domain with a family of such wavelets, with as few scales and offsets as possible. The Gabor wavelets do not form a finite basis, so perfect coverage cannot be achieved.

Many applications have been derived from Gabor analysis, however in the case where mostly orientation is sought, the Gabor wavelets may not be the best choice. Alternatives include Directional Filter Banks (DFB) wedge filters (Bamberger and Smith, 1992) with power-of-two resolution. These can be implemented with specific, efficient up- and down-sampling schemes using band-pass filter and a quincunx up/down sampling kernel . While the original wedge filters were designed primarily for image compression, more recent work (Truc et al., 2009) provided interesting improvements for image analysis. Quadrature derivative filters can



(a)



(b)

Figure 2.3: Example from Henriksen (2007): Gabor wavelets in the real domain, and the filter bank responses on a sample image with strong orientation and scale information.

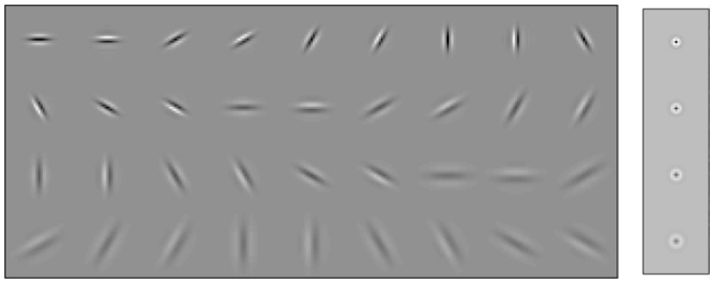


Figure 2.4: Filter bank from Malik et al. (2001):.

also be used (Malik et al., 2001). They essentially consist of perpendicular, rotated, Gaussian-convoluted derivative filters (i.e. DoG) with non-homogeneous variance (see Fig. 2.4. They are easy to compute and constitute a valuable alternative to Gabor filters.

We notice that most of these filter banks include a scale selection as well as orientation information. As we pointed out in the previous chapter in section 1.1.3, orientation and scale are linked. Another problem, which is related, is the that of deciding orientation at points were several objects are crossing, possibly of different scales. In the next section, we review some attempts to link orientation and space.

2.1.3 Orientation-space

Particularly in 2D image, but also sometime in 3D, objects overlap or cross each other due to projection effects or self-intersection. Also complex thin objects like vessel systems exhibit ambiguous areas like branches. Orientation-space was proposed by Chen and Hsu (1989) and in a short note by van Vliet and Verbeek (1995). The idea is fairly straightforward: given any method of providing local orientation φ (e.g. Gabor filters or otherwise) for a 2D image at (x, y), discretize and record this information in binary form in the 3D volume (x, y, φ), then label connected components, taking into account the fact that φ is periodic (see Fig. 2.5).

Illustrations and applications can be found in van Ginkel et al. (2001), including segmentation of overlapping objects, texture and fingerprint identifications. Orientation space was rediscovered independently by Chen et al. (1998, 2000) in 1998 and used in 2D cardio X-Ray imaging. It was also used by Van Ginkel et al. (1999) for curvature estimation. In Perona (1998), as similar concept was used to defined orientation scale-space for the diffusion filtering of orientation information, however orientation scale-space was not defined as a (4D) space, and indeed no particular representation was proposed, only an abstract space.

2.1.4 Wavelets

Scale-space as a linear theory stems from continuous isotropic diffusion, which is equivalent to Gaussian smoothing with a continuous parameter. It has been studied extensively, but is not the only way to consider multi-resolution approaches. In signal processing initially, the idea of wavelets with discrete resolution steps has been very fruitful (Mallat, 1999). Separable versions of the wavelet transforms have been adapted to imaging almost right away and used in a large variety of applications, from denoising to image coding. However, with regard to orientation analysis, classical separable wavelets are not very useful, because only a small number of orientations are used. Wavelet representations can be very sparse for many signals and some images, but not so much in the case of oriented textures, thin objects and contours. Several approaches have been proposed to deal with this situation.

2.1.4.1 Contourlets

One major objective of oriented wavelets is to be able to represent contours efficiently, unlike most wavelet bases. Contourlets, proposed by Do (2001); Do and Vetterli (2005), combine a classical Laplacian pyramid (Burt and Adelson, 1983) and a directional filter bank (DFB) similar to the one of Bamberger and Smith (1992) described in section 2.1.2.2. The basic scheme is illustrated in Fig. 2.6

For contourlets, the following holds:

- With both perfect reconstruction low-pass and directional filter banks, then the contourlet transform achieves perfect reconstruction. It constitutes thus a frame;
- If both LP and DFB are orthogonal, then the frame is tight with bounds equal to 1;
- The redundancy factor of the frame is less than 4/3;
- Using finite impulse response (FIR) filters, the complexity of the decomposition is $O(N)$, for images with N pixels.

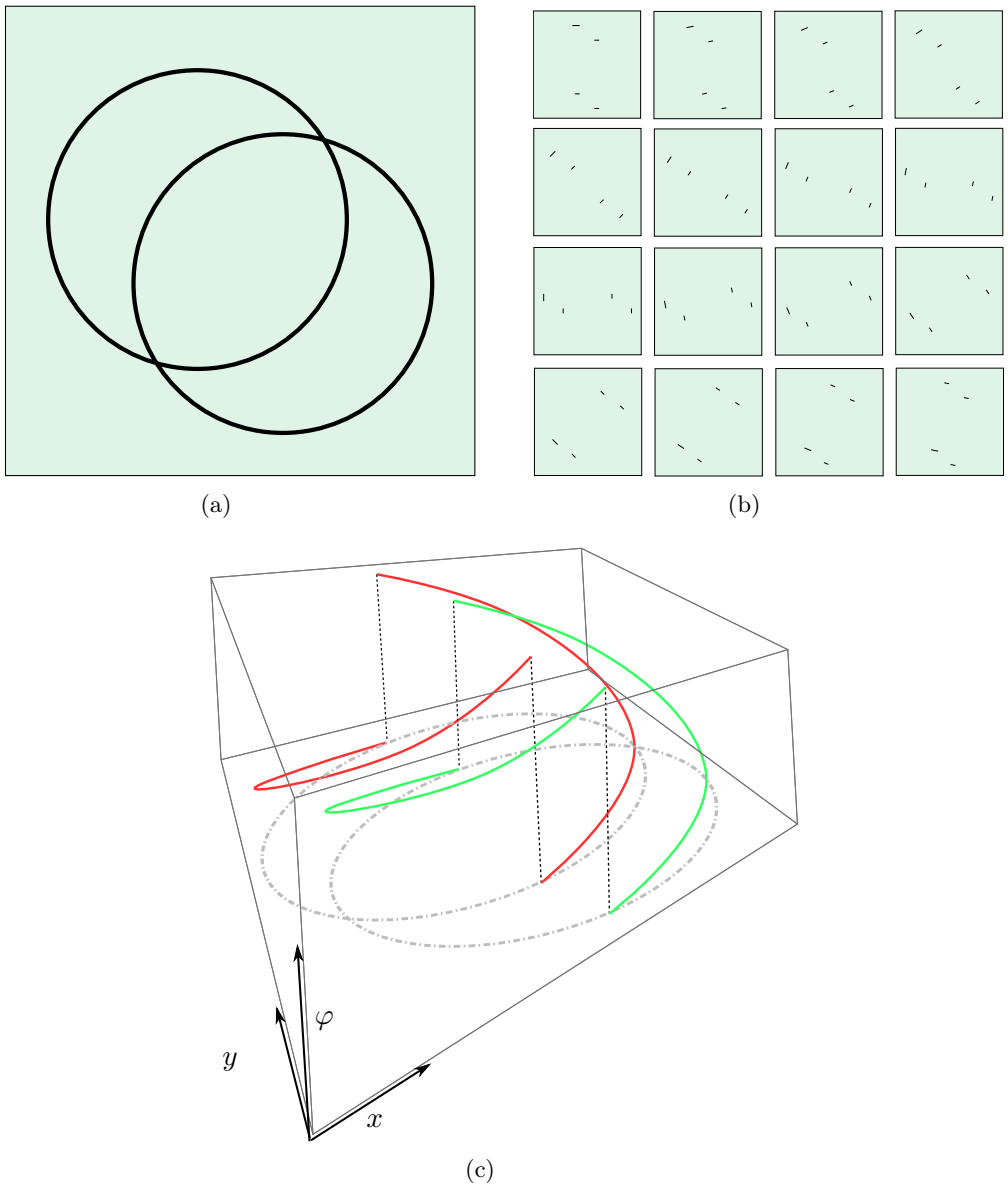


Figure 2.5: Two overlapping circles in orientation-space: in (a) two overlapping circles; in (b) the response of the orientation filter ; in (c) the labelled content of orientation-space, where the two circles appear as separate entities.

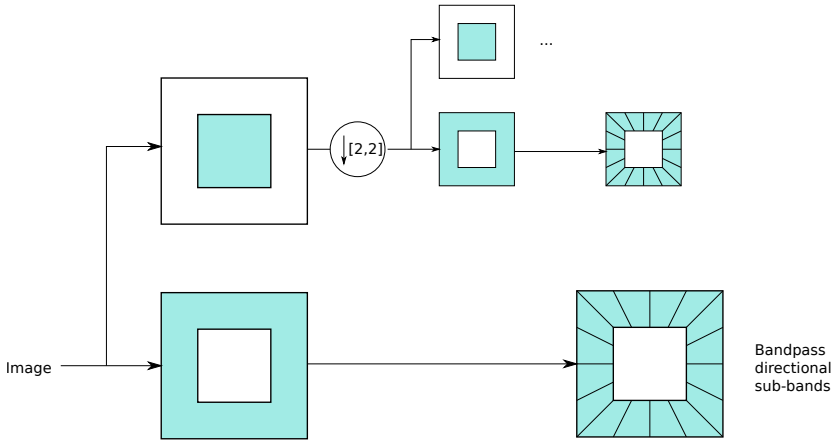
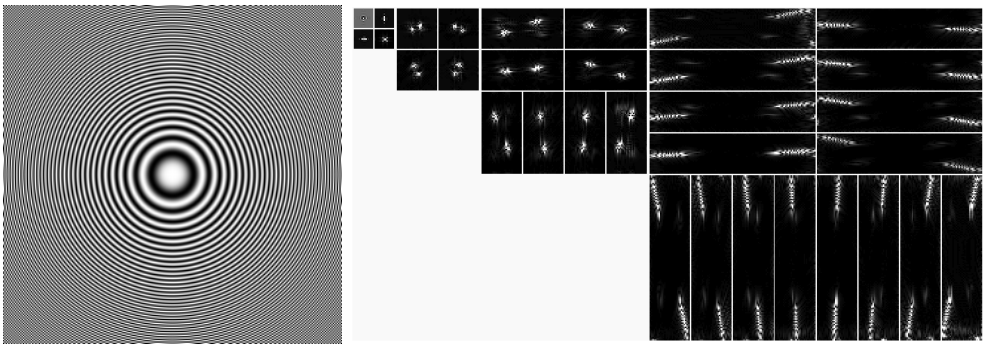


Figure 2.6: Contourlet filter bank: a multi-scale Laplacian pyramid decomposes the input image in octave bands, then a directional filter bank is applied to each bandpass channel.



(a) (b)

Figure 2.7: Example of contourlet decomposition on the "zone plates" test image.

An example of contourlet decomposition of an image with strongly oriented elements is given in Fig. 2.7. Notice that coefficient response is not necessarily oriented with the contours of the initial image. Nonetheless, for image compression and denoising, contourlets have been shown conclusively to perform better than separable wavelet bases in several applications (Villegas et al., 2008; Shan et al., 2009; Hiremath et al., 2011).

2.1.4.2 Ridgelets

Another attempt to define oriented wavelet are the fairly straightforward *ridgelets* (Candes, 1998; Candès and Donoho, 1999). The continuous ridgelet transform is defined in the following way: Let $\psi : \mathbb{R} \longrightarrow \mathbb{R}$ be a smooth univariate function satisfying the following wavelet admissibility condition:

$$C_\psi = \int_{\mathbb{R}} \frac{|\Psi(\omega)|^2}{|\omega|^2} \mathrm{d}\omega < +\infty, \tag{2.16}$$

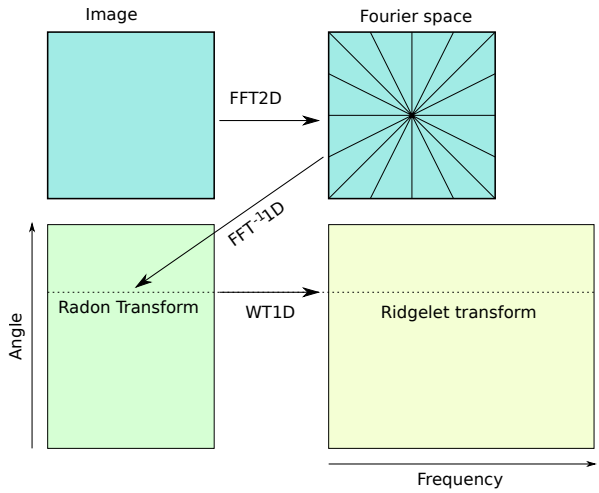


Figure 2.8: A conceptual implementation of the Ridgelet transform: images are sampled along centered rays in Fourier space. Radon slices are then transformed by the 1D wavelet.

where $\Psi(\omega)$ is the Fourier transform of ψ. Note that this implies $\Psi(0) = \int \psi(x)dx = 0$. We also assume this wavelet to be normalized:

$$\int_{\mathbb{R}} \frac{|\Psi(\omega)|}{\omega^2} \mathrm{d}\omega = 1. \tag{2.17}$$

Then, for each scale $a > 0$, each position $b \in \mathbb{R}$ and orientation $\theta \in [0, 2\pi)$, we define the *ridgelet* $\psi_{a,b,\theta}$ of $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ as:

$$\psi_{a,b,\theta}(\mathbf{x}) = \frac{1}{\sqrt{a}} \psi((x_1 \cos\theta + x_2 \sin\theta - b)/a). \tag{2.18}$$

A ridgelet is constant along the line $x_1 \cos\theta + x_2 \sin\theta = C$ with C constant. In the direction perpendicular to this line, it is a wavelet function. Ridgelet coefficients are defined by

$$\mathcal{R}_f(a, b, \theta) \equiv \langle f, \psi_{a,b,\theta} \rangle = \int_{\mathbb{R}^2} f(x) \overline{\psi_{a,b,\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{2.19}$$

The exact reconstruction is given by:

$$f(\mathbf{x}) = \int_0^{2\pi} \int_{\mathbb{R}} \int_0^{+\infty} \frac{1}{4\pi a^3} \mathcal{R}_f(a, b, \theta) \psi_{a,b,\theta}(\mathbf{x}) \mathrm{d}a \mathrm{d}b \mathrm{d}\theta \tag{2.20}$$

Ridgelets can be viewed as wavelet analysis in the Radon domain. The Radon transform in polar coordinates $(\rho, \theta) \in \mathbb{R} \times [0, 2\pi)$ is given by

$$Rf(\theta, \rho) = \int_{\mathbb{R}^2} f(x_1, x_2) \delta(x_1 \cos\theta + x_2 \cos\theta - \rho) \mathrm{d}x_1 \mathrm{d}x_2, \tag{2.21}$$

with δ the Dirac distribution.

The ridgelet transform is then the application of a 1D wavelet transform to the slices of the Radon transform, when θ is held constant and ρ varies. Therefore, a strategy to compute a ridgelet transform is first to compute a Radon transform on f and then a wavelet transform

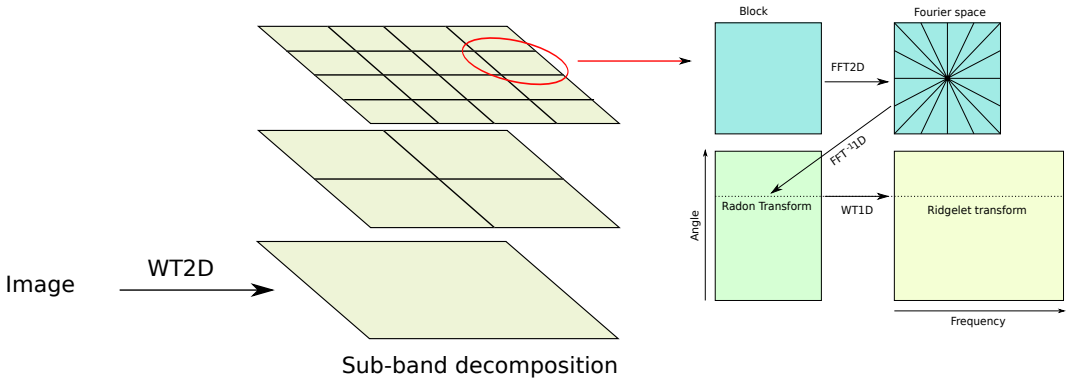


Figure 2.9: A conceptual implementation of the first-generation curvelet transform. The original image is decomposed into sub-bands by a 2D separable wavelet transform. Each sub-band is spatially partitioned, and each block analyzed by the ridgelet transform.

on the slices (see Fig. 2.8). However, since images are not continuous, a discretization strategy must be proposed. Several practical transforms are described and compared in Fadili et al. (2007).

The influence of ridgelets has been more theoretical than practical (indeed the entire thesis of Candès shows no experiments at all), since the underlying model assumes straight boundaries across the whole image. However some experiments with denoising shows that they perform adequately (Chen and Kégl, 2007), especially with a complex wavelet in the 1D transform (Chen and Kégl, 2007), or when the model corresponds to the image (Zhang et al., 2008).

2.1.4.3 Curvelets

Since the assumption of straight boundaries over the whole image is too strong, an interesting development is to use the ridgelet transform not on the whole image, but locally, on image sub-blocks. A natural idea is to use the multi-resolution bandpass decomposition of a separable wavelet transform to compute the sub-blocks. This led to the first generation *curvelet* transform (Donoho and Duncan, 2000; Starck et al., 2002), illustrated on Fig. 2.9.

However these curvelets are complex both to describe and compute, and highly redundant. This led Candès and Donoho in 2003 to propose a different construction, termed second-generation curvelets. These G2 curvelets are functions with vanishing moments in a given direction α like wavelets, but an elongated support in the direction $\alpha + \frac{\pi}{2}$, using different scaling factor along their width and length. Their frequency support is a wedge in Fourier space, constructed by the product of a radial window with an angular window. More precisely, a continuous curvelet atom, with scale $s > 0$, orientation $\theta \in [0, 2\pi)$, and position $\mathbf{y} \in [0, 1]^2$ is defined by:

$$\psi_{s,\mathbf{y},\theta}(\mathbf{x}) = \psi_s(R_\theta^{-1}(\mathbf{x} - \mathbf{y})), \tag{2.22}$$

with $\psi_s(\mathbf{x}) \approx 2^{-3j/4}\psi(x_1/\sqrt{s}, x_2/s)$, which is close to a parabolic stretch of a curvelet function ψ, with vanishing moments in the vertical direction. R_θ is the rotation operator of angle θ. At the given scale s, this curvelet atom is a thin object oriented in the direction θ, with an envelope is a ridge of length $1/\sqrt{s}$ and width s. It satisfies therefore a scaling property

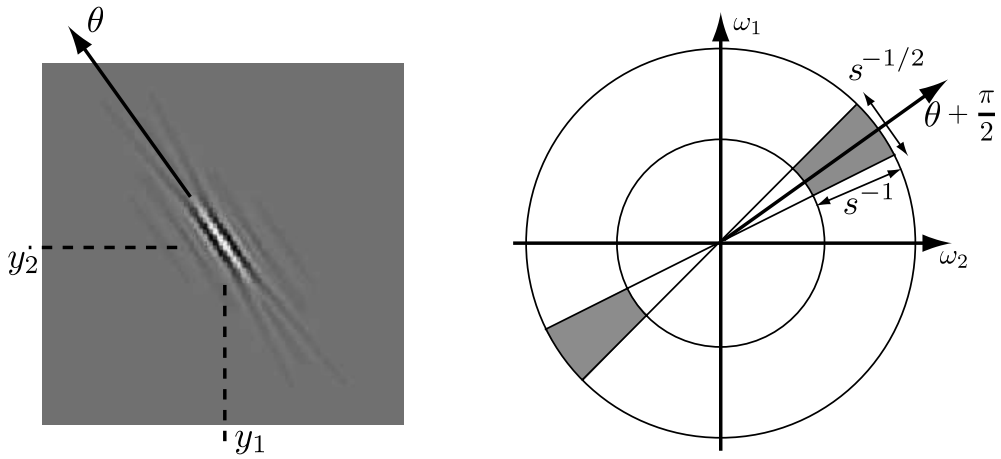


Figure 2.10: A second-generation curvelet $\psi_{s,\mathbf{y},\theta}(\mathbf{x})$. On the right, the frequency support of $\Psi_{s,\mathbf{y},\theta}(\omega)$ (from (Jacques et al., 2011))

of width = length^2. The continuous curvelet transform defines the set of inner products $\langle f, \psi_{s,\mathbf{y},\theta}(.)\rangle$ for all $(s, \mathbf{y}, \theta)$. A non-trivial design of ψ allows conservation of energy and a reconstruction formula (Candes and Donoho, 2003).

The continuous definition can be sampled to defined a wavelet frame, in such a way that curvelet parameters are sampled using an increasing number of orientations at finer scales, covering the whole frequency domain. The discrete curvelet transform also replaces radial windows by square ones and rotations with shears.

Curvelets have been shown to exhibit good properties with respect to denoising and compression of natural images.

2.1.4.4 Bandelets

These approaches depend on the notion of contour, which is not very well defined. Contrary to the previous approaches, *bandelets* (or bandlets) attempt to reduce the redundancy of orthogonal wavelet coefficients. They work through an operation called "bandeletization", which involves a segmentation of the wavelet domain. Similarly to curvelets, there has been at least two generations of bandelets and the literature can be confusing. This short summary takes as basis Mallat (2009, chap. 12). Further work related to bandelets can be found in Le Pennec and Mallat (2005); Mallat and Peyré (2007); Peyré and Mallat (2008). Some Matlab code used to be publicly available but is unfortunately no longer so at the time of writing.

The idea of bandelets is take advantage of the image geometric information, in particular of regularity alongside edges. This is implemented by a directional wavelet transform applied directly over orthogonal wavelet coefficients. This is called a "bandeletization" of wavelet coefficients. The directional wavelets are defined through a geometric approximation model. At each scale, and for each direction (typically three directions for orthogonal wavelets), the array of wavelet coefficients is divided into squares of varying size, following a quadtree structure encoding the non-regular regions (i.e. the regions with significant coefficients). In regular regions, coefficients are small and no further processing is required. In regions where a

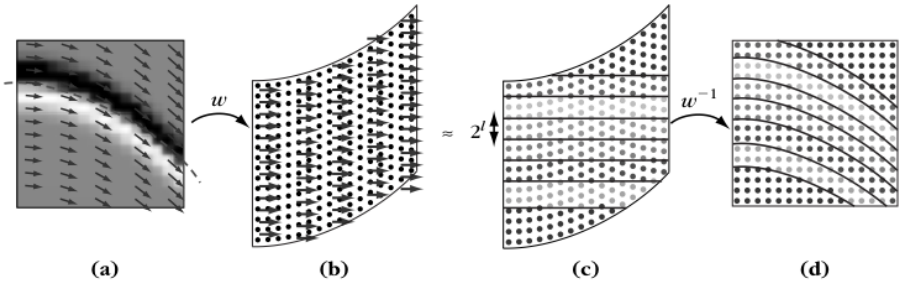


Figure 2.11: Geometric flow and derived directional wavelets (from Mallat (2009, p.635). in (a) the geometric flow, in (b) the warping procedure, in (c) wavelet encoding and (d) inverse warping.

clear direction cannot be found, these coefficients are not processed either, on the assumption that those are relatively rare. Near edges and oriented structures, a *geometric flow* is defined. This flow is a vector field which is locally parallel to itself either vertically or horizontally, providing the direction where the orthogonal wavelet coefficients are the most regular (the lines of the flow itself need not be aligned either horizontally or vertically). Wavelets coefficients are then *warped* in the direction perpendicular to the main flow lines, so that the flow lines themselves become either horizontal or vertical. Directional wavelets are then constructed from these warped coefficients, which are no longer located on grid points. Alpert wavelets (1993), suitable for non-uniform sampling grid, are used to represent the coefficients along the flow lines. The inverse warp provides the directional wavelets. The procedure is illustrated in Fig. 2.11.

Bandelets inherit a lot of regularity from the families of wavelets they use. Constructing an bandelet basis is for instance feasible. The idea of bandelets is quite original and seems promising, however a freely available reference implementation is lacking, and so evaluation is in fact difficult.

2.1.4.5 Other Xlets

Other wavelets with geometric or orientation information have been proposed, for instance various families of steerable wavelets (Simoncelli et al., 1992), the dual-tree complex wavelet transform (Kingsbury, 2001; Selesnick et al., 2005), and others. They combine orthogonal wavelet transforms with ideas presented earlier in this section (respectively steerable filters and Gabor wavelets) and have many interesting properties, however their discussion is somewhat redundant with what has already been presented.

Many more details, properties and illustrations regarding oriented wavelets can be found in the following documents: Fadili et al. (2007), Mallat (2009, chap. 5), Jacques et al. (2011).

2.1.5 Non-local filtering and patches

Most of the ideas presented so far, applicable to the filtering of thin objects, consist essentially of detecting directions in which objects vary as little as possible, in order to exploit this information in filter designs, e.g. by integrating in this direction only. Another idea is to compare regions which are broadly self-similar, and integrate over these regions. This is the idea behind *non-local means* (Buades et al., 2005).

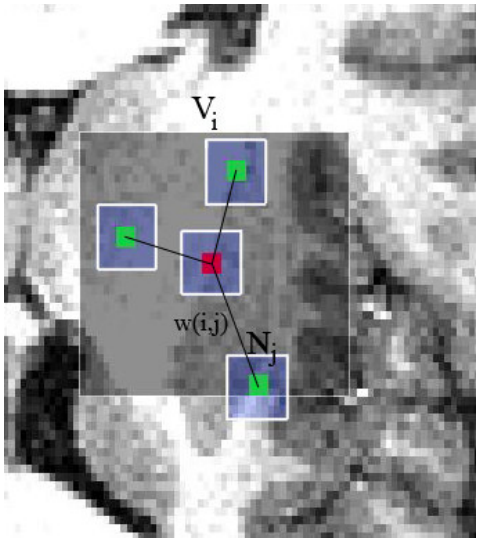


Figure 2.12: The principle behind non-local means: the denoised pixel at position i (in red) is a weighted average of the voxels j according to the similarity between patches around N_i and N_j, where N_j is restricted to the search volume V_i (figure Coupé et al. (2008a)).

Let Ω be the domain of a noisy image I_0, Let I be the denoised image, expressed in this fashion:

$$I(\mathbf{x}) = \frac{1}{\int_\Omega w(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{y}} \int_\Omega w(\mathbf{x},\mathbf{y}) I_0 \mathrm{d}\mathbf{y}, \tag{2.23}$$

where $\mathbf{x}$ is the local position, $\mathbf{y}$ all the non-local positions in the image, and w is expressed by

$$w(\mathbf{x},\mathbf{y}) = \exp\left(-\int_\Omega \frac{1}{h^2} G_\sigma(\mathbf{z}) |I_0(\mathbf{x}+\mathbf{z}) - I_0(\mathbf{y}+\mathbf{z})|^2 \mathrm{d}\mathbf{z}\right), \tag{2.24}$$

where G_σ is a Gaussian function of standard deviation σ and h is a tuning parameter. The term $\frac{1}{\int_\Omega w(\mathbf{x},\mathbf{y})}$ is a normalization factor. The obtained value in the denoised image at location $\mathbf{x}$ is simply a weighted average of all the other points $\mathbf{y}$ in the image. The weight is calculated as a Gaussian-weighted sum of squared difference between the regions around $\mathbf{x}$ (fixed for a given $\mathbf{x}$) and $\mathbf{y}$ (moving). In practical terms, this means that points in the entire image will be averaged together (i.e. denoised) if their neighborhoods are similar.

In practice this algorithm is too costly, indeed if n is the number of pixels in the image, then performing the averaging around each pixel requires n^2 differences, hence its complexity is in the order of $O(n^3)$, which is unworkable, not to mention useless, as soon as n is greater than 128×128, say. To make the algorithm tractable, we define a neighboring window N (or patches) related to σ and a search window V. We compute the sum of square differences between patches only in the search window around each pixel, as illustrated in Fig. 2.12. To further reduce the cost, only a small subset of similar windows linked to each pixel may be kept, prompting a so-called *patch-based* approach.

Since computing sum of squared differences is a window-summable operation, and since the computing of patch differences is highly parallelizable, this has led to some interesting implementations, for instance by Darbon et al. (2008), as well as GPUs (Liu et al., 2008;

Huang et al., 2009). A non-local denoising demo has even been part of the NVidia distribution of CUDA for several years (Kharlamov and Podlozhnyuk, 2007; Nickolls et al., 2008).

The non-local means approach delivers very good denoising results on natural images, and has inspired many authors to include non-local approaches in their research. Non-local patches have been proposed in various contexts, for instance used in variational regularization terms (Gilboa and Osher, 2008; Bresson and Chan, 2008; Bresson, 2009; Peyré et al., 2011). Many patch similarity measures have been proposed, some using learning approaches (Mairal et al., 2009; Mairal, 2010). Links to wavelet analysis have also been proposed by Dabov et al. (2006, 2007). This combined approach of wavelet and patch-based regularization is the basis of the BM3D algorithm (Danielyan et al., 2012; Marc Lebrun, 2012) which is currently widely considered to be the state of the art in image denoising, along with several similar methods (Kervrann et al., 2007; Coupé et al., 2008a). Note, however, that this has not been confirmed in the particular case of thin objects.

In the context of thin objects filtering, NLM have been used in the context of 3D MRI (Coupé et al., 2008b). We will show other examples of applications of NLM-type methods in section 2.3.

2.2 Non-linear, morphological operators

Up to now we have broadly reviewed linear approaches. We focus this section on non-linear filters, particularly those based on mathematical morphology.

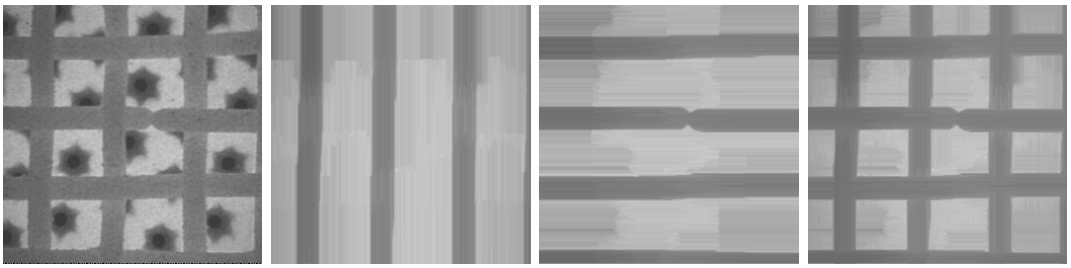
2.2.1 Operations with line segments as structuring elements

When seeking to filter thin objects, a common assumption is to consider that they are locally oriented and therefore, at some scale, they can be assimilated locally to a short line segment. A simple idea to filter such objects in the framework of mathematical morphology is to use a filter bank, in this case by making use of the well-known properties of openings and closings, namely that a supremum of openings is an opening; and an infimum of closing is a closing.

A natural structuring element in this case is a segment of a given length L. Since morphological[1] openings and closings yield the same result, it does not matter whether they are centered or not. The length L provides a parameter that offers a compromise between the effectiveness of the filter and the maximum curvature of the feature one seeks to keep. A straightforward, parallel implementation of this idea consist of generating a family of line segments, and simply performing the successive openings and closings, retaining respectively the supremum or the infimum, which can be built iteratively.

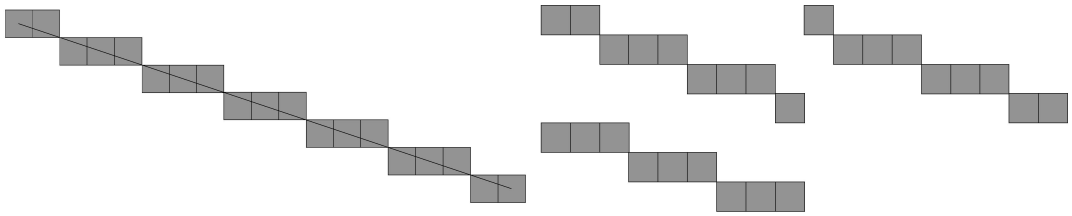
Using an effective algorithm for computing arbitrary erosions and dilations, such as in (Vandroogenbroeck and Talbot, 1996) and especially (Urbach and Wilkinson, 2008) may help, but the complexity of a single such openings/closings remains of the order of $O(L \times N)$ in the worst case, with N the number of pixels in the image. Assuming one samples D directions, one requires $O(L \times N \times D)$ operations. In order to cover all the useful angles, $D \approx \pi N$, and so the total complexity is $O(L^2 \times N)$. Nonetheless such algorithms were used with a small D (fewer than 10) in some applications. One of the earliest documentation for such an

[1] I.e. structuring element-based, as opposed to algebraic, which may not be the composition of an erosion and a dilation.



(a) (b) (c) (d)

Figure 2.13: The classical "burner" application showing a metal grid with gas burner behind (a). The objective here is to detect the grid, which can be performed by an infimum (d) of a vertical (b) and an horizontal (c) closing with a long and thin structuring element (100×1 *and its transpose)*



(a) (b)

Figure 2.14: The reason why algorithms using Van Herk-like recursive computation along oriented lines to perform openings with oriented segments are not translation-invariant. In (a) we show a Bresenham discrete line. In (b) its decomposition into successive segments 9 pixel long. All segments indeed have the same length, however the arrangement of successive pixels is not the same from one segment to the next.

instance is in the `Micromorph` example manual (Gratin, 1989) with the "burners" application (see Fig. 2.13).

In order to improve speed, in 1992 Van Herk (1992) proposed an algorithm to compute MM operators along vertical or horizontal lines in constant time irrespective of L. A straightforward extension of the Van Herk algorithm is to consider sloped discrete lines, for instance Bresenham lines (1965), which then yields operators with arbitrarily oriented line segments. This lowers the complexity to $O(L \times D)$ (Soille et al., 1996). Improvements of the basic Van Herk scheme proposed by Gil and Kimmel (2002) and then Vandroogenbroek and Buckley (2005) helped improving the speed even further. However, the recursive approach common to all these algorithm loses the translation invariance property of the straightforward parallel implementation. We will look into this problem in section 2.3.1.1, however Fig. 2.14 explains this problem. Using a recursive computation along a line segment amounts to using different segments as structuring elements depending on the origin location.

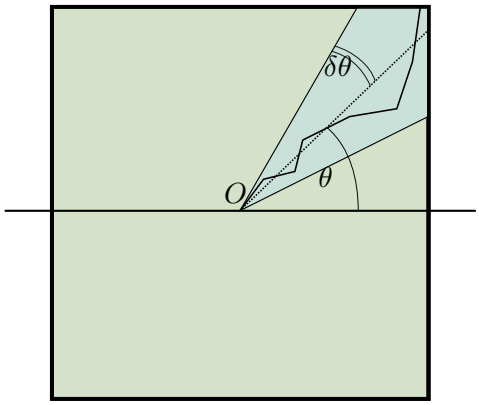


Figure 2.15: Vincent's local paths: a local minimal path is computed from the center of a window at O to the border of the window, constrained in a semi-cone oriented along θ and of aperture $\delta\theta$.

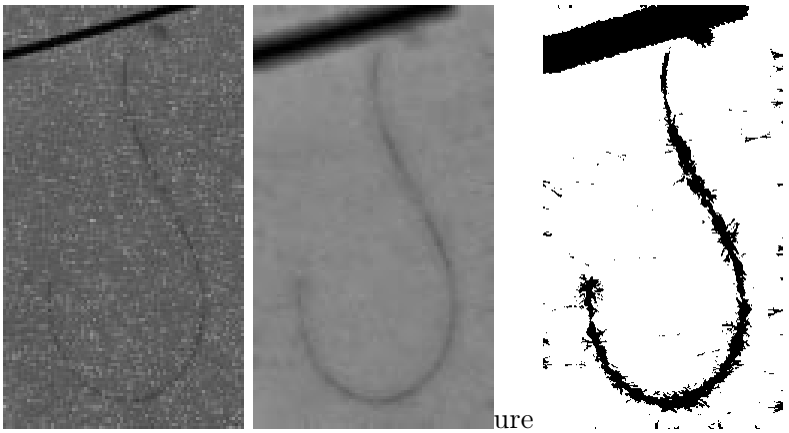
2.2.2 Binary-only approaches

Finding the orientation of binary thin objects is a problem in itself. They can be handled as a grey-level object of course, but it is sometime useful to exploit the binary nature of the data. For instance, Kurdy and Jeulin in (1989), as well as Tuzikov et al. (1992) proposed to use rotated pairs of points as structuring elements and covariogram-like approaches to find the orientation of binary objects. Following a suggestion by Beucher, in Soille and Talbot (1998) we proposed to take as local orientation at a point $x \in A$ measure the direction of the longest segment fitting in A at x. (Altendorf and Jeulin, 2009) have proposed to use the combined information from several distance transforms to compute local orientation in 2D and 3D. This is an extension of a classical approach consisting in computing the gradient of the distance transform of a binary object, although the latter is not very precise. In text processing, the de-skewing of a scanned page requires the computation of a global orientation, which can be achieved very cheaply by projection and histogram (Bloomberg and Vincent, 2010). Since it is fairly unusual to benefit from purely binary data as input, these approaches suffer from the need to obtain a segmentation first. As orientation is a strong cue for segmentation, these methods are often hampered by a chicken-and-egg situation.

2.2.3 Local minimal path

Minimum-cost paths can be computed in images at a relatively low cost, by considering typically a restriction of the local connectivity graph (Dijkstra, 1959; Rosen and Vincent, 1994). Since paths embed some local orientation information, an idea is be to consider families of paths going through a point. Vincent (1998) proposed to compute shortest paths constrained by oriented (semi-)cones (or wedges), as shown on Fig. 2.15. In spite of their apparent complexity, these shortest paths can be efficiently computed in scan order in parallel. Paths can be more or less constrained depending on the aperture of the considered cones. Given a sufficiently small aperture, one can increase the angular resolution arbitrarily, however more constrained cones tend to look more like line segments, losing some of the specificity of this approach.

Local shortest paths can be used in several fashions. A simple way is to record at each



(a) (b) (c)

Figure 2.16: Fibre segmentation with local shortest paths. (a) A thin fibre on a noisy background; notice the disconnections. (b) Image of the smallest length of local shortest paths, using cones of aperture 15 degrees. (c) threshold of image (b), detecting the fibre.

cone origin the smallest of all the (weighted) length of all the constrained local paths. It is also useful to record the orientation of the cone from which this path originates, as well as other information about all the other paths from different cones, such as the highest, or the average length. In this way, it is possible to distinguish cone origins coming from a flat regions (in which most paths will be of equal length) vs. an oriented region, where some paths might cost significantly more or less than the average path. Vincent used the smallest constrained path as a new metric on the image, from which to derive further processing.

2.2.4 Connected morphological filters

Work of Breen and Jones, Work of M. Wilkinson, also some work at the CMM for retina angiography in 2D. This is quite important !

2.2.5 Geodesic paths and path voting

Discrete paths can be computed using Dijkstra-like methods, however more regular versions of the same concept can be derived from the Fast-Marching Method and similar (Tsitsiklis, 1995; Sethian, 1999a). Such paths are sometime termed *geodesic paths*. Significant work has been derived from path computations in 2D and 3D (Cohen and Kimmel, 1997; Ardon and Cohen, 2006; Cohen and Deschamps, 2007; Benmansour and Cohen, 2009, 2011).

An interesting approach to the filtering and segmentation of thin objects is geodesic voting (Rouchdy and Cohen, 2008). The idea is to seed the computation of many geodesic paths throughout the image, which tend naturally to converge to thin structures. In flat regions only a few geodesic paths are present, whereas in areas where oriented objects are present, the density of paths tend to increase. A measure based on the density of such path leads to useful metrics for the detection of thin objects, although shading effects sometimes affect the results (Rouchdy and Cohen, 2009).

2.2.6 Other approaches

Many other, more ad-hoc, approaches have been proposed to detect thin structures. They tend to depend on properties that are not always easy to generalize outside of a few applications. One exception is the oriented median Graf and Wörn (2007), which uses directional information from the structure tensor. A 1D median filter is then applied instead of a more common diffusion approach. 1D median filtering in orthogonal directions is sometimes used to filter oriented structures that are not uniformly dark or white (Sun et al., 2006). Unstructured graph-based approaches have been used when thin objects are sufficiently wide, yet to noisy to be detected as one feature (Viero and Jeulin, 1995).

Importantly, topological information, which is a strong cue but is not always easy to deal with, has been used in edge and thin object linking (Shih and Cheng, 2004) as well as thin surfaces segmentation (Dokladal et al., 1999; Cardoso et al., 2011). One critical aspect is that topological information is a fragile global property, whereas most image analysis is local and affected by noise. In some instances, as above, when the whole dataset is visible (e.g. whole organ like the liver or the brain), this can prove very useful.

2.2.7 Demosaicing

Demosaicing is a very specific problem, which has however been widely studied because of the relatively recent ubiquity of colour sensor arrays. CFAs are a sensible and cost-effective solution to colour sensing and have contributed widely to the miniaturization and democratization of digital cameras. It is interesting to note that even in recent cameras, the most commonly used CFA is the original Bayer contribution, dating from the 1970s (E.Bayer, 1976). In spite of its many flaws and incorrect assumptions, so much work has been proposed to correct the artifacts that it generates, that the quality of the resulting images is effectively "good enough". Meanwhile the patent has lapsed and camera manufacturers have been able to reap the rewards of this research, as well as of a good part of the research put forward by the academic community on this problem.

A very good literature review of many CFA demosaicing approaches was written by Gunturk et al. for the IEEE Signal Processing Magazine (2005) in 2005. Since then some progress has been done, but all the issues are well described in this article.

2.2.8 Discussion

Although it cannot be claimed that the previous sections constitute a complete coverage of the processing and analysis of thin objects in the imaging literature, it is hopefully representative of the variety of efforts researchers have spend studying these objects. Also, at this stage it should be obvious that even though thin objects are not the most studied features in image analysis, their processing is important. Virtually all areas of image processing have devoted some effort to their particular cases, which are unfortunately neither unified nor simple. As a result the solution proposed are not all satisfactory. Taking as example the wavelet domain, there does not exist as yet a method of representing edges or oriented textures which is fully satisfactory. Arguably the most promising approach, bandelets, is not a fully developed technology. Other linear filters, in spite of a large variety of useful properties, either propose something that is too local or too global. Less effort has been extended to non-linear filters, as the community working in this area is indeed smaller, however there has been some promising approaches, which deserve to be extended. This is the topic of the next section.

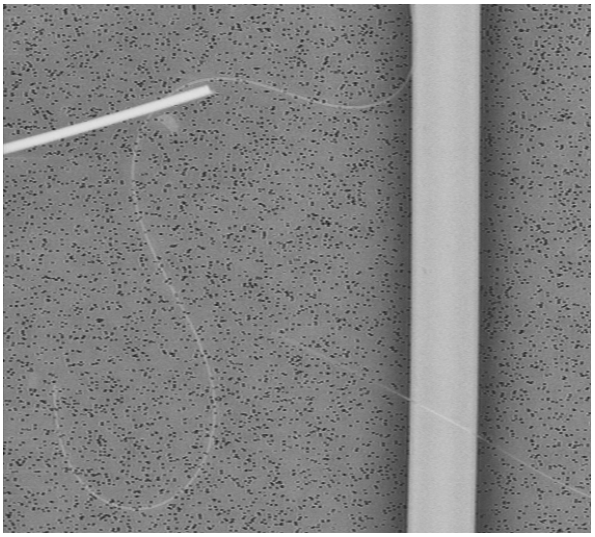


Figure 2.17: An image of thin man-made vitreous fibres, including thin ones.

2.3 Contributions to thin objects filtering

In this section, we present and discuss contributions to the domain of thin object filtering and segmentation.

2.3.1 Segment-based operators

Since my original training is in mathematical morphology, it should come as no surprise that some of my contributions were made in this framework. My original Ph.D work involved the filtering and segmentation of man-made vitreous fibres, which appear in 2D as a superposition of multiscale, thin, oriented objects, as on Fig. 2.17

In spite of the simplicity of the problem: measuring the diameter of all the visible fibres, it is in fact difficult. No fibre can be missed and they shouldn't be counted more than once. Fibres can be very thin indeed, their diameter reaching down to the highest resolution of the microscope. The risk of missing a fibre is therefore high. Since they are long, and corrupted with noise, as well as overlapping, the problem is non-local, the risk of separation is high, and so the risk of multiple counting is high as well. In addition, the thinnest fibres are in fact those of highest interest. This provided motivation for many years of research, and indeed I do not consider the problem as solved even today.

2.3.1.1 Faster, less noise sensitive, and translation-invariant morphological operators with line segments

As noted in section 2.2.1, one way to filter thin objects is to consider them locally straight, and fast operators for MM operators using line segments have been proposed in the literature. However there were two remaining problems with these approaches: i) the lost property of translation-invariance and ii) the noise sensitivity.

The translation-invariance property of filtering operators is always important, but especially so for thin objects. Indeed, it is common in applications to have several similar objects

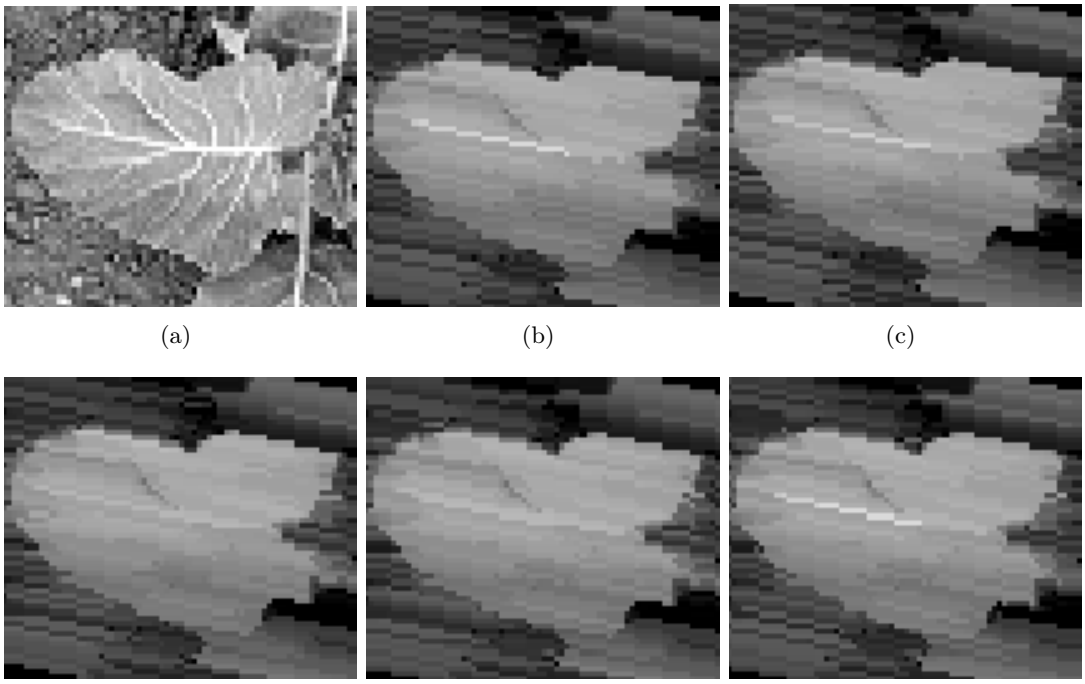


Figure 2.18: Image of cauliflowers leaves (a), opened by Bresenham line segments with the same orientation and the same length (15), but of different origin (b-f). The results can be startlingly different.

to filter in the same image. If the operator does not behave in exactly the same way throughout the image, then parameter optimization can become impossible. For thin objects, the difference might be quite noticeable, as shown on Fig. 2.18, and can lead to unreliable results.

Noise sensitivity is a somewhat different issue, since it is a feature of the operator itself. Indeed the geometrical interpretation of an opening (or a closing) helps explaining the problem. Assuming any set F, a morphological opening operator γ_S with structuring element S will not filter it out completely it, i.e. $\gamma_s(F) \neq \emptyset$ if and only if $S \subset F$. However, if F is line-like, i.e. thin and long, then S must be as well, to discriminate F from other objects. However, noise becomes more and more problematic as the length of S increases. Indeed, in the limit a single noise pixel along the length of F might ensure that S will not fit into it, and so γ_S will filter it out. As shown on Fig. 2.19, in the presence of noise, only segment with small length can be used, longer segments erase all features. However segments with small lengths are not able to eliminate the noise.

2.3.1.2 Restoring the translation invariance

The situation can be described in the following: we want to keep using recursive algorithms to compute segment-based MM operators because of the speed benefits, but we do not want to lose the translation invariance property.

In (Soille and Talbot, 2001) we propose two solutions, which share a common part. The common part is as follows: First we restrict ourselves to orientations that can be expressed as

(a) (b) (c)

Figure 2.19: Line filtering sensitivity in the presence of noise: in (a) a synthetic image of line-like objects. In (b), the same image corrupted by salt-and-pepper noise. In (c), the result of the union of openings by arbitrarily oriented segments of length 5.

a rational number. This is not a drawback concerning visual and quantitative quality of the output, yet it ensures that the number of operations we need to compute is finite (and indeed, relatively small). Both solution involve computing a number of Bresenham lines for any given orientation, indeed, all the Bresenham lines that have the same slope but a different origin.

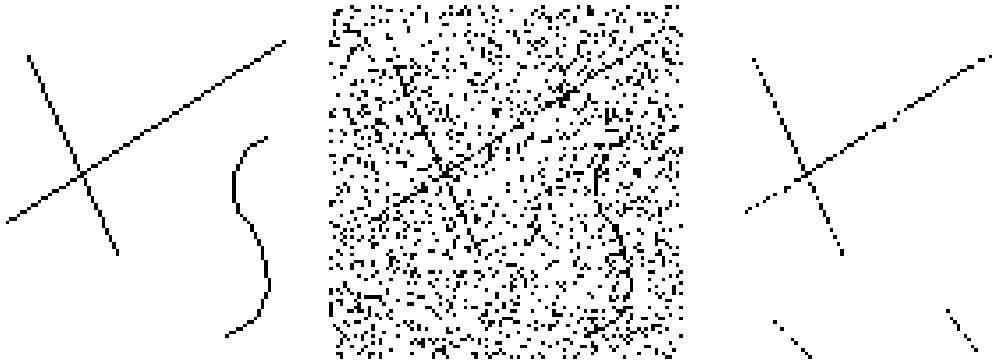
Denoting $\psi_{BL_{\lambda_i}(dx,dy)}$ the non translation-invariant (TI) morphological operator performed along a Bresenham line of length λ, of origin i, and orientation vector (dx, dy), with $k = \max|dx|, |dy|$ and $i \in 1, \ldots, k$. We denote $\psi_{L_{\lambda_i}(dx,dy)}$ the TI equivalent. We show the following property:

$$\bigvee_{i=1}^{k} \gamma_{BL_{\lambda_i}(dx,dy)} = \bigvee_{i=1}^{k} \gamma_{L_{\lambda_i}(dx,dy)} \tag{2.25}$$

$$\bigwedge_{i=1}^{k} \varphi_{BL_{\lambda_i}(dx,dy)} = \bigwedge_{i=1}^{k} \varphi_{L_{\lambda_i}(dx,dy)} \tag{2.26}$$

In other words, since these operators are respectively openings and closings on both sides of the equal signs, it is possible to generate TI operators from non-TI ones, at the expense of having to consider a larger family of operators. On the other hand, we argue that considering this larger family yields a much finer final operator, which is better able to preserve thin details, and we do extract significant benefits from using this larger family beyond the translation-invariance only.

To compute the actual operator, this is where the two solutions diverge. We can either use the van Herk recursive formula or any of its variants, or we can use a sliding window combined as proposed in (Gil and Werman, 1993) and extended to arbitrary shapes and combined with an histogram search in (Vandroogenbroeck and Talbot, 1996). While the recursive formula is relatively straightforward, we also proposed a translation-invariant way to visit successive pixels along a particular oriented line, which is compatible with the sliding-window algorithm. This sliding-window algorithm is particularly interesting, because it allows us to combine morphological and rank filters, which is extremely useful for noise sensitivity.



(a) (b) (c)

Figure 2.20: Line filtering with rank-max opening: in (a) a synthetic image of line-like objects. In (b), the same image corrupted by salt-and-pepper noise. In (c), the result of a union of rank-max openings using segments as structuring elements. Compare with Fig. 2.19.

2.3.1.3 Improving noise sensitivity

Noise sensitivity of morphological operators is a serious problem, which has been well illustrated for instance by Heijmans (1996) and undermines the effectiveness of the framework in some applications. For instance, given a set A and an opening γ_B by a structuring element B. We assume $\gamma_B(A) = A$ with a noise-free A. Irrespective of the size of A and B, if A is corrupted by a single noise pixel to form A', it could well happen that $\gamma_B(A') = \emptyset$. Various strategies have been developed to cope with this problem, giving rise to the theory of morphological filtering (Serra and Vincent, 1992), which depending on the context may be more or less flexible (Bloch and Maitre, 1994), computationally intensive or effective (Meyer, 2004). However one family of morphological filters, solves this particular problem very effectively (Soille, 2002):

Rank order filters are at the basis of very powerful openings and closings called rank-max openings and rank-min closings. These operations have been originally proposed by Ronse (1986), also in Ronse (1988). The idea of rank-max operators is instead of using a plain discrete structuring element B whose cardinal number equals n pixels, a variable structuring element is used instead. Taking as example the opening, the rank-max opening consists of taking the supremum of the morphological openings by all possible subsets $B' \subseteq B$ with cardinal $k, 1 \leq k \leq n$.

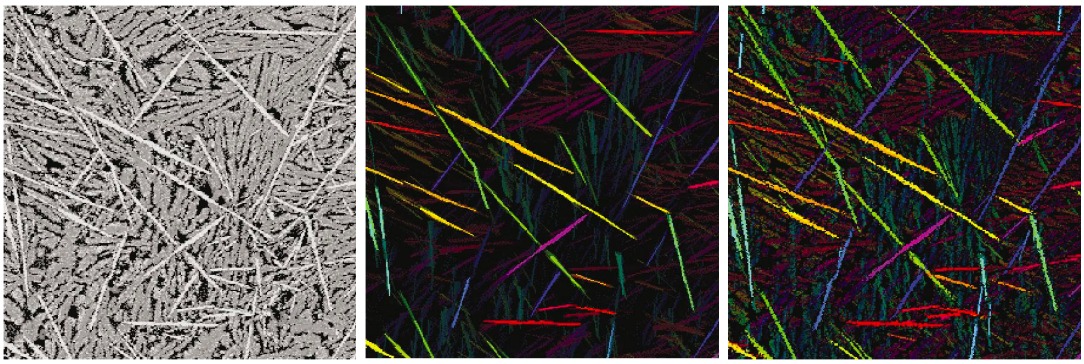
Denoting $\gamma_{B,k}$ the rank-max opening with structuring element B and parameter k, we have

$$\gamma_{B,k} = \bigvee \gamma_{B'}, B' \subseteq B \text{ and } \operatorname{card}(B') = k, \tag{2.27}$$

with $1 \leq k \leq \operatorname{card}(B)$. Computing this operator with this definition is prohibitive, fortunately, Ronse showed that the following expression is equivalent:

$$\gamma_{B,k} = \mathrm{id} \wedge (\delta_B \zeta_{B,n-k+1}), \tag{2.28}$$

Where $\wedge$ is the pointwise infimum, δ_B is the adjunct dilation to erosion ε_B, and $\zeta_{B,l}$ is the rank operator using l as rank. We see that when k is 1, $\zeta_{B,n}$ reduces to the max filter, i.e.



(a) (b) (c)

Figure 2.21: On an image of fibres in a microprobe (a). Comparison between translation invariant opening (b) and rank-max opening (c). Colours indicate orientation.

the dilation operator. Since the dilation is extensive, $\gamma_{B,1}$ reduces to identity through the pointwise minimum operator. On the other hand, if k is n, $\zeta_{B,1}$ reduces to the min filter and the rank-max operator is a classical opening. In between these values, the rank-max opening has the properties described above, which have the effect of reducing the noise sensitivity of morphological operators.

Since noise-sensitivity is even more important for thin objects, it is useful to be able to compute rank-max operators with segment-based morphological operators. Fortunately this is possible using our TI operators described in the section above, particularly the sliding-windows implementation, which is suitable for any rank operator. In Fig. 2.20, we illustrate the capacity of a union of rank-max opening with line segments as structuring elements to restore the thin and straight parts of an image, with some false positives.

In Fig. 2.21 we show a real example of fibres imaged in an X-ray microprobe. Orientation is calculated pointwise by the opening that yields the highest response. We see that this information is useful, naturally highly correlated, and that the rank-max version yields a less noise-sensitive result, as expected.

The combination of translation-invariance and noise robustness properties make these radial operators useful in a number of applications. Some will be shown at the end of this chapter.

2.3.1.4 Other remarks

The supremum of openings with line segments is sometimes called the *radial opening*, and complement operator: the infimum of closings with line segments is sometimes called *radial closing*. From these several other interesting operators can be built, including oriented granulometries, orientation histograms, segmentation methods using orientation, and oriented top-hat.

An important point is the efficiency of these operators. In spite of the larger family of SEs that must be considered to retain the translation invariance, the TI radial operators still keep a low complexity. By considering the Farey sequence (1816) to find out which individual operator to perform, their number is kept to a minimum.

These important details are described bySoille and Talbot (1998, 2001). This latter paper

is reprinted in this document as Appendix D. Please refer to it for more details.

2.3.2 Operations with paths as structuring elements

In Fig. 2.20, we show the result of a morphological, rank-max radial opening. While the straight elements of the original image are sufficiently well restored (with some false positives), the curved thin object is completely lost. The reason for this, of course, is that it is not sufficiently locally straight to fit with the model of the radial opening. To cope with this sort of objects, we need a larger family of structuring elements: some that are thin, but not necessarily locally straight.

One idea, inspired by the approach of Vincent (1998) it to use families of paths constrained in such a way that they are always subset of a cone at every vertex. However, if we consider for instance a cone of aperture $\frac{\pi}{2}$ and an 8-connected graph, at each vertex three edges are admissible for path continuation, and so if the path is of length L, the total number of admissible paths is 3^L, which quickly becomes a very large number.

Nonetheless, since morphological operators can be computed recursively, we now show we can actually define and compute such operators.

2.3.2.1 Graphs and path operators

In this section we recall the definition of path openings and closing, and we expand their use to include morphological profiles. In this and the following section, all definitions are given for the binary image space $\mathcal{P}(E)$. The results can be generalised to the space of grey-scale images $\mathrm{Fun}(E, T)$ by means of the thresholding theorem for flat morphological operators (Heijmans, 1994, Chapter 11). Path-based morphological operators were first described in Buckley and Talbot (2000), and again with a much better formalism in Heijmans et al. (2004).

Let E be the image domain endowed with a binary *adjacency* relation $x \mapsto y$, meaning that that there is an edge going from x to y. In general, the relation '$\mapsto$' is non-symmetric, which means that the graph given by the vertices E and the adjacency relation $\mapsto$ is a *directed graph*. If $x \mapsto y$, we call y a successor of x and x a predecessor of y. Using the adjacency relation we can define a dilation on $\mathcal{P}(E)$ by writing

$$\delta(\{x\}) = \{y \in E \mid x \mapsto y\}\,. \tag{2.29}$$

In other words, the dilation of a subset $X \subseteq E$ comprises all points which have a predecessor in X. These concepts are illustrated on Fig. 2.22. Here b_1, b_2, b_3 are successors of a and $\delta(\{a\}) = \{b_1, b_2, b_3\}$. Furthermore, a_1, a_2, a_3 are the predecessors of b and $\breve{\delta}(\{b\}) = \{a_1, a_2, a_3\}$.

The L-tuple $\mathbf{a} = (a_1, a_2, \ldots, a_L)$ is called a *δ-path of length L* if $a_k \mapsto a_{k+1}$, or equivalently, if

$$a_{k+1} \in \delta(\{a_k\})\,, \text{ for } k = 1, 2, \ldots, L-1\,. \tag{2.30}$$

Note that $\mathbf{a} = (a_1, a_2, \ldots, a_L)$ is a δ-path of length L if and only if the reverse path $\breve{\mathbf{a}} = (a_L, a_{L-1}, \ldots, a_1)$ is a $\breve{\delta}$-path of length L. Given a path $\mathbf{a}$ in E, we denote by $\sigma(\mathbf{a})$ the set of its elements:

$$\sigma(a_1, a_2, \ldots, a_L) = \{a_1, a_2, \ldots, a_L\}\,. \tag{2.31}$$

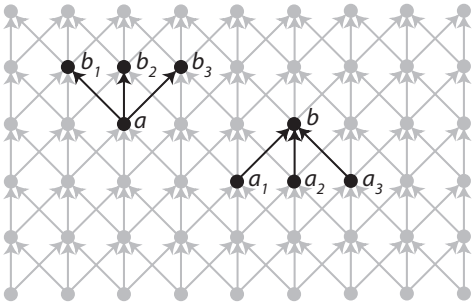


Figure 2.22: b_1, b_2, b_3 are successors of a and a_1, a_2, a_3 are the predecessors of b.

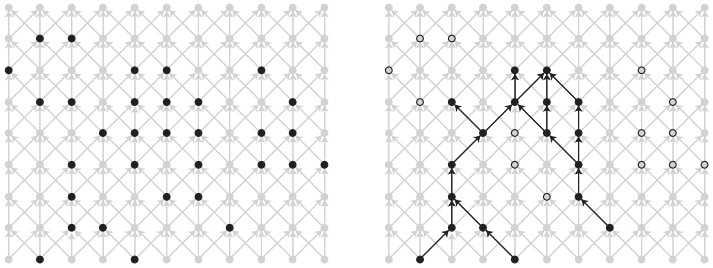


Figure 2.23: A set $X \subseteq E$ (black points on the left) and its opening $\alpha_6(X)$ (black points on the right). The points on the right with a white center are discarded by the transform.

We denote the set of all δ-paths of length L by Π_L. The set of δ-paths of length L contained in a subset X of E is denoted by $\Pi_L(X)$, i.e.,

$$\Pi_L(X) = \{\mathbf{a} \in \Pi_L \mid \sigma(\mathbf{a}) \subseteq X\}\,, \tag{2.32}$$

We define the operator $\alpha_L(X)$ as the union of all paths of length L contained in X:

$$\alpha_L(X) = \bigcup \{\sigma(\mathbf{a}) \mid \mathbf{a} \in \Pi_L(X)\}\,. \tag{2.33}$$

The transform α_L has all the properties of an algebraic opening (Heijmans et al., 2005), and we call it the *path-opening*. Conversely, *path-closings* are defined by straightforward complementation (i.e. exchanging foreground and background). We illustrate the result of a simple path opening on Fig. 2.23.

In (Heijmans et al., 2005) we provide a recursive formula for computing paths operators with length parameter $k+1$ from the path operators with length k, and an associated polynomial complexity algorithm.

Path openings and closings are dependent on the notion of graph connectivity. In order to be useful, this connectivity should reflect the kind of paths that the application requires. Examples of useful graphs are those that define cones oriented in the principal directions of the grid, as shown in Fig. 2.24. Path openings and closings in these graphs are those that retain paths that at each point fit in a $\frac{\pi}{2}$ angle cone, oriented in a principal direction. Combinations by supremum (for openings) and infimum (for closings) make it possible to retain paths oriented in all possible directions just using these four adjacencies.

Employed in this manner, path openings and closings can be used to retain features that are locally oriented but not necessarily perfectly straight. Path openings and closings can be

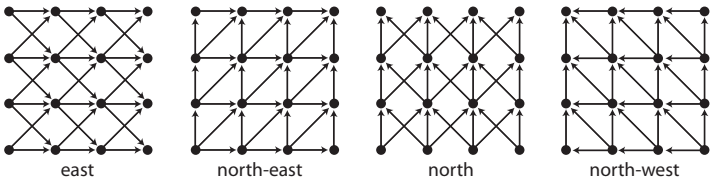


Figure 2.24: Four different adjacencies defining overlapping $\frac{\pi}{2}$ cones.

implemented efficiently both in the binary and grey-level cases with a linear time complexity with respect to L using a decomposition algorithm Heijmans et al. (2005). Improved algorithms were proposed in Appleton and Talbot (2005a); Talbot and Appleton (2007). In terms of efficiency, these path-based operators are about as fast as radial operators for the same dimensions (segment length vs. path length). In terms of translation-invariance, path-based operators inherit this property from the graph they use. Since the graphs in Fig. 2.24 are TI, so are the path-based operators they are based upon. Paths with smaller aperture are typically not TI, so composition by Sup/Inf can be performed to restore this property as with the segments-based operators.

2.3.2.2 Path-based operators and noise sensitivity

Similarly to segment-based openings, operators with paths are noise-sensitive to some degree. Because paths may be sufficiently flexible, depending on the adjacency graph that is used, they can "route" around noise pixels to some degree, however it is still useful to add a measure of robustness. This is achieved in a different way to segment-based operators, because a recursive way to compute rank filters does not exist to our knowledge, and so formula 2.28 may not be employed in the path context.

Instead, we have to rely on a more "brute-force" approach.

The path-opening $\alpha_L(X)$ of a set X comprises the union of all length-L paths contained inside X. We can relax this condition by demanding that only k out of L vertices of the path lie inside X, thus yielding a so-called *incomplete path-opening* $\alpha_L^k(X)$. We present a formal definition below.

Define $\Pi_L^k(X)$ as the collection of length-L paths in E which contain at least k points inside X:

$$\Pi_L^k(X) = \{\mathbf{a} \in \Pi_L \mid |\sigma(\mathbf{a}) \cap X| \geq k\}\,. \tag{2.34}$$

Note that this definition only makes sense for $0 \leq k \leq L$, and that

$$\Pi_L(X) = \Pi_L^L(X) \subseteq \Pi_L^{L-1}(X) \subseteq \cdots \subseteq \Pi_L^1(X) \subseteq \Pi_L^0(X) = \Pi_L\,. \tag{2.35}$$

We define the *incomplete path-opening* as

$$\alpha_L^k(X) = \bigcup\{\sigma(\mathbf{a}) \cap X \mid \mathbf{a} \in \Pi_L^k(X)\}\,. \tag{2.36}$$

Note that it is obvious that

$$\alpha_L^L \leq \alpha_L^{L-1} \leq \cdots \leq \alpha_L^0\,, \tag{2.37}$$

and that

$$\alpha_L^L = \alpha_L \text{ and } \alpha_L^0(X) = \{x \in X \mid \Lambda(x) \geq L\}\,, \tag{2.38}$$

(a) (b) (c)

Figure 2.25: Line filtering with incomplete path openings: in (a) a synthetic image of line-like objects. In (b), the same image corrupted by salt-and-pepper noise. In (c), the result of a incomplete path opening with length 15 and gap 2. Compare with Fig. 2.19 and 2.20. Here we recover both the lines and the curve.

where $\Lambda(x)$ is the length of the longest paths that contains x. Putting

$$\bar{E}_L = \{x \in E \mid \Lambda(x) \geq L\}\,. \tag{2.39}$$

we get that

$$\alpha_L^0(X) = X \cap \bar{E}_L\,. \tag{2.40}$$

Furthermore, we define

$$\psi_L^k(X) = \{\alpha_1 \mid \mathbf{a} \in \Pi_L^k(X)\}\,. \tag{2.41}$$

We have

$$\psi_L^L = \psi_L \text{ and } \psi_L^0(X) = \{x \in E \mid \breve{\lambda}(x) \geq L\} = \bigcup_{k \geq L} \breve{E}_k\,. \tag{2.42}$$

Using the the convention that $\psi_L^k \equiv \varnothing$ if $k > L$, it is possible to express ψ_{L+1}^{k+1} in terms of ψ_L^{k+1} and ψ_L^k (Heijmans et al., 2005), and from there to express a polynomial-complexity algorithm.

In its binary form, the above recursion expresses the fact that the path operator can be computed by a propagation operation first following the underlying connectivity graph, and then in the reverse way. This enough to provide a workable general purpose implementation. This was first performed by M. Buckley prior to the year 2000 using threshold decomposition. In Talbot and Appleton (2007), however, we proposed a much more efficient, ordered algorithm.

This ordered algorithm, considers threshold levels from lowest to highest, and keeps in data structures the paths that it encounters, building them recursively with the same formulas from (Heijmans et al., 2005). We show that the complexity of this approach is linear in terms of gaps pixels (i.e. a path operator allowing p gaps will take $O(p)$ longer than a path with zero gaps).

Table 2.2: Some reference algorithms for the segmentation and classification of blood vessels

Method	Accuracy (standard deviation)
2nd observer	0.9473 (0.0048)
Path operators	0.9445 (0.0084)
Zana *et al.*	0.9377 (0.0077)
Jiang *et al.*	0.9212 (0.0072)
Martinez-Perez *et al.*	0.9181 (0.0240)
Chaudhuri *et al.*	0.8773 (0.0232)

On Fig. 2.25, we show the result of using an incomplete path opening on our image of thin objects corrupted by noise. We find that the incomplete path opening is capable of eliminating all the noise which is sufficiently far away from the features of interest, both straight and curved. From this result, finding the structure of interest is relatively easy (for instance, perform an isotropic closing followed by thinning). We are not aware of any other transform with similar characteristics.

Many more details about path operators are given in Appendix E, which is a reprint of (Heijmans et al., 2005).

2.3.2.3 Applications in 2D

Path operators, being relatively new, have not yet been used extensively. They are useful for filtering textures and objects that are long and thin. Some applications have included

- Hair detection and removal on skin lesions (Skladnev et al., 2004);
- Small glass and mineral fibre segmentation (Talbot et al., 2000);
- Neurite segmentation (Bischof et al., 2005);
- Nerve analysis on the cornea and non-destructive testing (Morard, 2012);
- Retina blood vessel analysis (**?**).

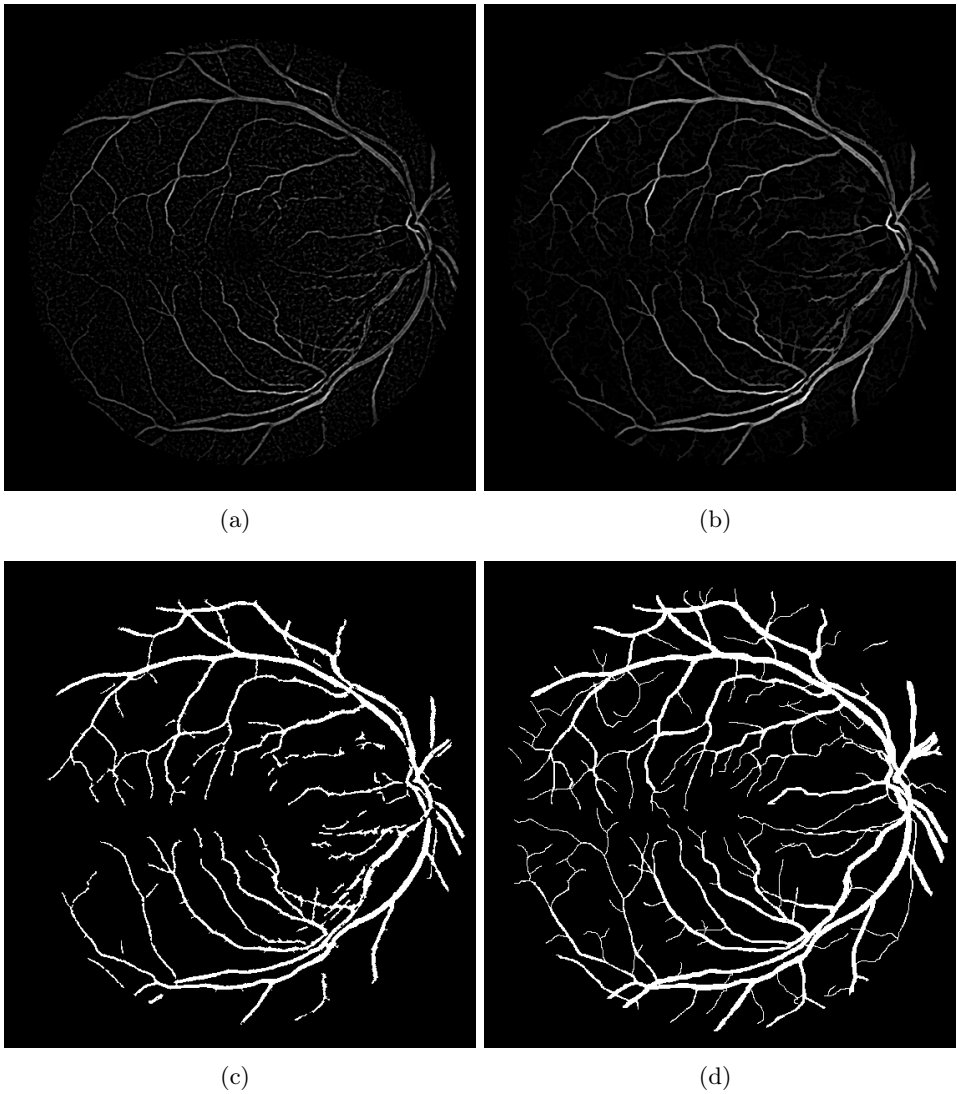
This latter topic is interesting because a limited public benchmark is available: the Digital Retinal Image for Vessel Extraction (DRIVE) at `http://www.isi.uu.nl/Research/Databases/DRIVE/` Staal et al. (2004), with 40 images and their associated ground truth. Some results are shown on Fig. 2.26.

The full method is described in Már Sigurdsson et al. (2013). This analysis is ongoing as of this writing, however based on early results, we have obtained the accuracy shown on Table 2.2.

The competing methods are referenced in Niemeijer et al. (2004). The results provided by path operators are very encouraging, however, as shown on Fig. 2.26, improvements are still possible and necessary.

2.3.2.4 Extensions to 3D

Extensions of path operators to 3D or more are not problematic from the theoretical point of view. Indeed, only a directed graph is necessary, which can easily be embedded in an



(a) (b)

(c) (d)

Figure 2.26: Eye fundus blood vessel analysis. (a) reference DoG ridge detection ; (b) path closing detection; (c) path closing classification; (d) ground truth.

arbitrary dimension lattice. However, from the point of view of the implementation, efficiency is important because the number of possible oriented graphs increases and neighborhood of any point becomes more complicated.

C. Luengo Hendriks was the first to propose a 3D implementation with some interesting improvements in (Hendriks, 2010), which he used and evaluated in an application for wood fibre analysis. However, this implementation does not include incomplete paths, which are very useful for noise robustness.

A different approach for path noise robustness was proposed by F. Cokelaer *et al.* (Cokelaer et al., 2012). Whereas incomplete path propose a global criterion for robustness (i.e. so many pixels may be noise over the whole given length of the path), robust path operators (RPO) propose a local criterion, by which any given succession of pixels may be noise, anywhere in the path. The proportion of noise pixels is not given, only how many such pixels in succession may be noise anywhere along the path, in arbitrary proportion. It turns out that the latter approach requires much less accounting, but provides in practice similar results.

The use of 3D path operators is being currently investigated for crack detection, brain and heart vessel segmentation. In particular alternatives to classical vesselness measures based on these operators are of interest. A freely available implementation of path operators is available at `http://hugues.zahlt.info/` as well as `http://pinkhq.com`. Cris Luengo also maintain a version as part of DIPlib at `http://www.diplib.org/`.

2.3.3 Other approaches

Many other approaches have been proposed for thin objects analysis. Performing a complete analysis and comparison would be highly desirable, but fall out of the scope of the current work.

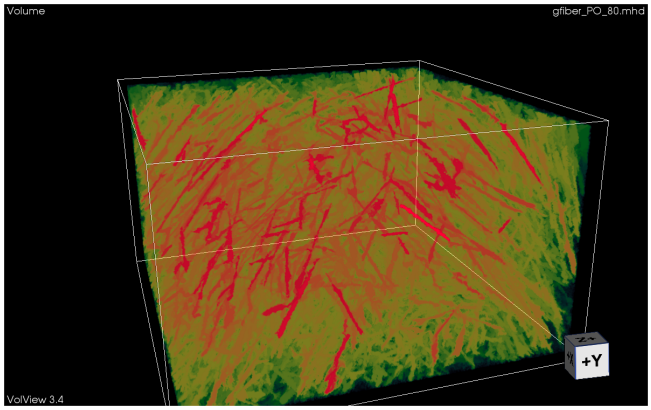
Among the approaches that we have investigated but that we do not have the space to present here are the combined linear-morphology approach of the Morpho-Hessian by Tankyevych et al. (2008, 2009), recently improved with non-local filtering in Nguyen et al. (2013).

We have also investigated shortest-path and fast-marching-based approaches, inspired by the seminal work of Cohen and Kimmel (1997), as well as more recent work by Benmansour and Cohen (2011). This has given rise to an efficient method for segmenting object based on closed contour detection (Appleton and Sun, 2003), providing an exact 2D solution to the Geodesic Active Contour model (Appleton and Talbot, 2005b), that we will present in the next part of this work.

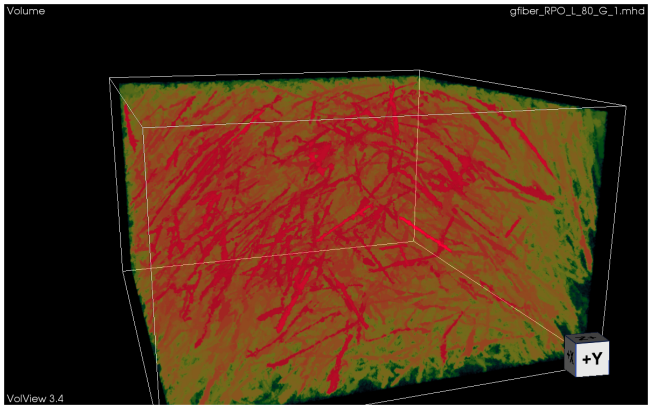
Recently, inspired by geodesic voting techniques introduced by Rouchdy and Cohen (2008, 2009), we have proposed an efficient 2D polygonal path voting technique (Bismuth et al., 2012), which is very efficient at finding very thin structures in noisy environments, as illustrated on Fig. 2.28.

2.4 Conclusion

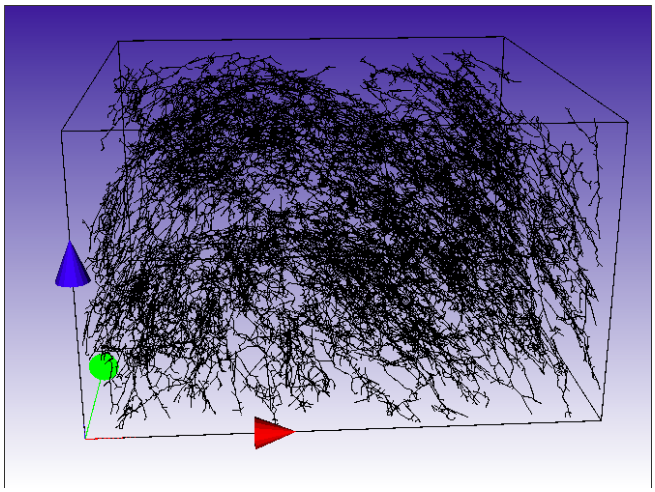
In this long chapter, we have first provided a survey of the state of the art regarding the filtering and segmentation/identification of thin objects in images, and we have outlined our contribution.



(a)



(b)



(c)

Figure 2.27: Path operators on 3D fibres: (a) complete operator; (b) robust operator; (c) classification. Results from Cokelaer (2013)

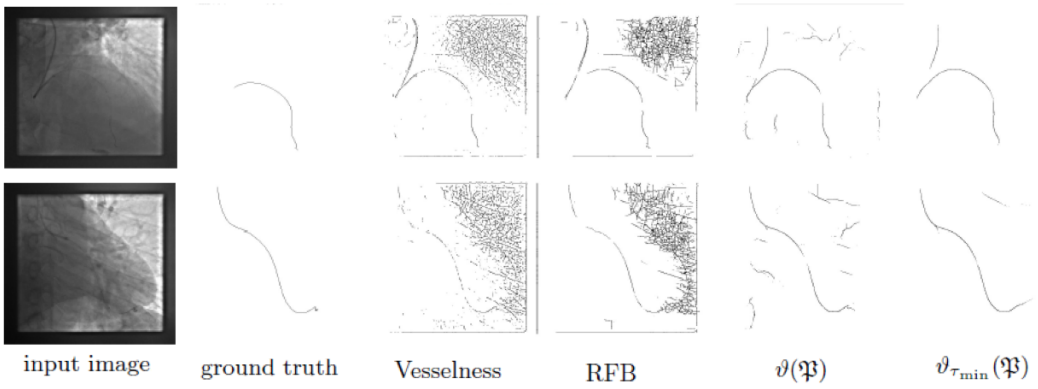


Figure 2.28: X-ray fluorescence guide-wire extraction. Vesselness, Rotated Filtered Banks and Polygonal path images, from Bismuth et al. (2012).

On the survey of thin object filtering While we cannot claim to have performed a *thorough* survey, one conclusion is that most methods are set in the linear framework. These include derivative-based filters, filter banks, steerable filters and of course a number of wavelet frames and bases. The objectives of these approaches were not unique, the filtering of thin objects being only one of the application. Other applications include in particular image denoising/restoration with the constraint of preserving contours, and image coding/sparse representation in the presence of arbitrarily oriented textures, thin objects and contours. In this context, it is probably fair to say that only a subset of this work concerns 3D or higher-dimension data. Exceptions include all the vesselness-oriented work using derivative-based filters. Filter banks become expensive to compute in 3D, steerable filters beyond the basic ones difficult to formulate. There exist some work on 3D wavelets but is as advanced and sophisticated as in 2D. 3D curve and surface wavelet representations in particular are all but absent.

In the non-linear domain, significant work has been based on shortest paths derived from the fast-marching method, for vessel identification in particular. In this area the work on surface identification has not been as convincing. Mathematical morphology work in this area has focused on the use of segments as structuring elements, based on the hypothesis that thin objects are at some scale straight enough. For filtering work, some elongation-based connected thinning operators have also been proposed. They have been used for instance for volume rendering, or in 2D angiography.

On our contribution Over the last few year, our work has focused on two areas: improving the segment-based morphological operators, and developing path-based operators. Segment-based operators were either slow or lost some important property like translation-invariance, which practically meant that they could not be guaranteed to preserve some thin objects, depending on their location in the image. We have shown how to reconcile the two, while also providing a measure of noise-robustness via a rank-max formulation. Nonetheless segment-based operators are fundamentally limited by their model, which supposes that at some scale thin objects are segment-like, which may not be true.

For this reason, we have developed the theory and practice of path-based morphological operators, which remove this assumption. These operators bridge the gap between segment-

based operators, which are very oriented and assume local regularity and connected operators like area-based openings and closings, or elongation-based thinnings which are much more isotropic. Path based operators can behave more like one or the other, depending on the underlying graph geometry and connectivity that is given. We have provided noise-resistant operators via incomplete paths and robust paths, and we have extended them to 3D, all the while providing robust and efficient algorithms.

We have also briefly outlined our contribution regarding combining linear and non-linear operators in the Morpho-Hessian filter for thin objects, the polygonal path image, and the link we have drawn between shortest-path computation and the segmentation of objects by the detection of their closed contour as a thin object. This allowed us to formulate a practical and exact solution to the well-known Geodesic Active Contour (GAC) problem. However interesting, this solution only works in 2D, which led us to investigate flow-based methods to solve the same problem in 3D or more. This is the topic of the next part, which will take us in a completely different domain.

Part II

Flow methods in image analysis and computer vision

Introduction

In this part we consider both discrete and continuous segmentation methods, particularly those based on flow simulations.

Maximum flow methods are a classical way of solving a particular class of linear programming problems, that are a subclass of transport methods. These methods are highly efficient and usually solved on directed graphs. Due to seminal work of Y. Boykov, O. Veksler, and R. Zabih (Boykov et al., 1998) to provide a general framework for formulating and solving useful and interesting problems in computer vision on graphs, applications in imaging have blossomed in this area.

We became interested in both discrete and continuous optimization methods soon after 2000 but somewhat independently of this particular work. At that time PDE-based methods (finite differences and level sets) were the standard and so we cast our work in this framework. More recently it has become clear that these methods can also be viewed in a convex optimization framework, which is in fact more versatile and flexible.

This part includes a review of seeded, or so-called interactive segmentation methods, a chapter on discrete methods and a chapter on continuous methods.

Chapter 3

Seeded segmentation methods for medical image analysis

Segmentation is one of the key tools in medical image analysis. The objective of segmentation is to provide technique for reliable, fast and effective organ delineation. While traditionally, particularly in computer vision, segmentation is seen as an early vision tool used for subsequent recognition, in medical imaging the opposite is often true. Recognition can be performed interactively by clinicians or automatically using robust techniques, while the objective of segmentation technique is to precisely delineate contours and surfaces. This can lead to effective techniques known as "intelligent scissors" in 2D and their equivalent in 3D.

This chapter is divided as follows. Section 3.1 starts off with a more "philosophical" section that sets the background for this study. We argue for a segmentation context where high-level knowledge, object information and segmentation method are all separate.

In section 3.2, we survey in some details a number of segmentation methods that are well-suited to image analysis, in particular of medical images. We illustrate, make some comparisons and some recommendations.

In section 3.3 we introduce very recent methods that unify many popular discrete segmentation methods and we introduce a new one. In section 3.4 we give some remarks about recent advances in seeded, globally optimal active contour methods that are of interest for this study.

In section 3.5 we compare all presented methods qualitatively and give some discussion. We then conclude and give some indications for future work.

3.1 The need for seed-driven segmentation

Segmentation is a fundamental operation in computer vision and image analysis. It consists of identifying regions of interests in images that are semantically consistent. Practically, this may mean finding individual white blood cells amongst red blood cells; identifying tumors in lungs; computing the 4D hyper-surface of a beating heart, and so on.

Applications of segmentation methods are numerous. Being able to reliably and readily characterize organs and objects allows practitioners to measure them, count them and identify them. Many images analysis problems begin by a segmentation step, and so this step conditions the quality of the end results. Speed and ease of use are essential to clinical practice.

This has been known for quite some time, and so *numerous* segmentation methods have been proposed in the literature (Pham et al., 2000). However, segmentation is a difficult

problem. It usually requires high-level knowledge about the objects under study. In fact, semantically consistent, high-quality segmentation in general is a problem that is indistinguishable from strong Artificial Intelligence and has probably no exact or even generally agreeable solution. In medical imaging, experts often disagree amongst themselves on the placement of the 2D contours of normal organs, not to mention lesions. In 3D, obtaining expert opinion is typically difficult, and almost impossible if the object under study is thin, noisy and convoluted, such as in the case of vascular systems. At any rate, segmentation is, even for humans, a difficult, time-consuming and error-prone procedure.

3.1.1 Image analysis and computer vision

Segmentation can be studied from many angles. In computer vision, the segmentation task is often seen as a low-level operation, which consists of separating an arbitrary scene into reasonably alike components (such as regions that are consistent in terms of color, texture and so on). The task of grouping such component into semantic objects is considered a different task altogether. In contrast, in image analysis, segmentation is a high-level task that embeds high-level knowledge about the object.

This methodological difference is due to the application field. In computer vision, the objective of segmentation (and grouping) is to recognize objects in an arbitrary scene, such as persons, walls, doors, sky, etc. This is obviously extremely difficult for a computer, because of the generality of the context, although humans do generally manage it quite well. In contrast, in image analysis, the task is often to *precisely* delineate some objects sought in a particular setting known in advance. It might be for instance to find the contours of lungs in an X-Ray photograph.

The segmentation task in image analysis is still a difficult problem, but not to the same extent as in the general vision case. In contrast to the vision case, experts might agree that a lesion is present on a person's skin, but may disagree on its exact contours (Menzies et al., 1996). Here the problem is that the boundary between normal skin and lesion might be objectively difficult to specify. In addition, sometimes there does exist an object with a definite physical contour (such as the inner volume of the left ventricle of the heart). However, imaging modalities may be corrupted by noise and partial volume effects to an extent that delineating the precise contours of this physical object in an image is also objectively difficult.

3.1.2 Objects are semantically consistent

However, in spite of these difficulty, we may assume that, up to some ambiguity, an object (organ, lesion, etc) may still be specified somehow. This means that semantically, an object possess some consistency. When we point at a particular area on an image, we expect to be, again with some fuzziness, either inside or outside the object.

This leads us to the realization that there must exist some mathematical indicator function that say whether we are in or out of the object with high probability. This indicator function can be considered like a series of constraints, or labels. They are sometimes called *seeds* or *markers*, as they provide starting points for a segmentation procedures, and they mark where objects are and are not.

In addition, a *metric* that expresses the consistency of the object is likely to exist. A gradient on this metric may therefore provide object contour information. Contours may be weak in places where there is some uncertainty, but we assume they are not weak everywhere

(else we have an ambiguity problem, and our segmentation cannot be precise). The metric may simply be the image intensity or color, but it may express other information like consistency of texture for instance. Even though this metric may contain many descriptive elements (as a vector of descriptors for instance), we assume that we are still able to compute a gradient on this metric (Sagiv et al., 2006).

This is the reason why many segmentation methods focus on contours, which are essentially discontinuities in the metric. Those that focus on region do so by defining and utilizing some consistency metric, which is the same problem expressed differently.

The next, and final step for segmentation, is the actual contour placement, which is equivalent to object delineation. This step can be considered as an optimization problem, and this is the step on which segmentation methods in the literature focus the most. We will say more about this in section 3.2 listing some image segmentation categories.

3.1.3 A separation of powers

In summary, to achieve segmentation in the analysis framework, we need three ingredients: (1) an indicator function that say whether we are in or out of the object of interest ; (2) a metric from which we may derive contour information and (3) an optimization method for placing the contour accurately.

To achieve accuracy, we need flexibility and robustness. Some have argued that it is useful to treat these three steps separately. This was first described in Meyer and Beucher (1990)) as the *morphological* method, but this also called by others *interactive* or *seeded* segmentation (Grady, 2006b). In this context, this does not mean that user interaction is required, only that object identification is provided by some means, and contour extraction is provided separately by a segmentation operator.

The first ingredient, the object identification, or our indicator function, is of course essential and it is frustrating to be obliged to only write here "some means". Accurate content identification can simplify the requirements on the segmentation operator greatly. Unfortunately, the means in question for contents identification are problem-dependent and sometime difficult to publish, because they are often seen as *ad-hoc* and of limited interest beyond their immediate use in the problem at hand. Fortunately some journals allow such publications, e.g. the *Journal of Image Analysis and Stereology* and applications journals (e.g. *Journal of Microscopy*, materials, etc). There are also very few recent books on the matter, although some do exist (Najman and Talbot, 2010; Dougherty and Lotufo, 2003). Software libraries are also important and not many are freely available for training, although the situation is improving.

Also whereas in computer vision a fully automated solution is required, in medical imaging a semi-automated method might be enough. In bio-medical imaging, a large number of objects are typically measured (such as cells, organelles, etc), and a fully-automated method is often desirable. However, in medical imaging, typically a relatively small number of patients is being monitored, treated or surveyed, and so human-guided segmentation can be sufficient. The objective of the segmentation method in this context is to provide reasonable contours quickly, that can be adjusted easily by an operator.

In this variety of contexts, is it possible to define precisely the segmentation problem? The answer is probably no, at this stage at least in image analysis research. However, it is possible to provide *formulations* of the problem anyway. While this may sound strange or even suspicious, the reason is that there exists a real need for automated or semi-automated

segmentation procedures for both image analysis and computer vision, and so solutions have been proposed. They can still be explained, compared and evaluated.

3.1.4 Desirable properties of seeded segmentation methods

We come to the first conclusion that to provide reliable and accurate results, we must rely on a segmentation procedure and not just an operator. Object identification and constraints analysis will set us in good stead to achieve our results, but not all segmentation operators are equivalent. We can list here some desirable properties of interactive segmentation operators.

- It is useful if the operator can be expressed in an energy or cost optimization formulation. In this fashion, it is amenable to existing optimization methods, which entails a number of benefits. Lowering the cost or the energy of the formulation can be done in several ways (e.g. continuous or discrete optimization), which results in different characteristics and compromises, say between memory resources and time. Optimization methods improve all the time through the work of researchers, and so our formulations will benefit too.
- It is desirable if the optimization formulation can provide a solution that is at least locally optimal, and if possible globally optimal, otherwise noise will almost certainly corrupt the result.
- The operator should be fast, and provide guaranteed convergence, because it will be most likely restarted several times, in order to adjust parameters. Together with this requirement, the ability to segment many objects at once is also desirable, otherwise the operator will need to be restarted as many time as there are objects in the image. This may not be a big problem if objects do not overlap and if bounding boxes can be drawn around them, because the operator can then be run only within the bounding box, but this is not the general case.
- The operator should be bias-free: e.g. with respect to objects size or to the discretization grid or with respect to initialization.
- The operator should be flexible: it is useful if it can be coupled with topology information for instance ; or with multi-scale information.
- It should be generic: not tied to particular data or image types.
- It should be easy to use. This in practice means possessing as few parameters as possible. Of course one can view constraints setting as an enormous parameter list, but this is the reason why we consider this step as separate.

Such a method certainly does not yet exist to our knowledge, but some might be considered to come close. We describe some of them in the next section.

3.2 A review of segmentation techniques

Here we list and detail some segmentations methods categories that are compatible with the image analysis viewpoint. We cannot hope to present a complete description of this field, but we hope to be helpful in presenting a few categories.

3.2.1 Pixel selection

Pixel selection is likely the oldest segmentation method. It consists of selecting pixels solely based on their values and irrespective of their spatial neighborhood. The simplest pixel selection method is the humble thresholding, where we select pixels that have a grey-level value greater or smaller than some value. This particular method is of course very crude, but used all the time nonetheless Multiple thresholding uses a range of values instead of a single value ; color and multi-spectral thresholding using vectors of values and not just scalars. By definition, all histogram-based methods for finding the parameters of the thresholding, including those that optimize a metric to achieve this (Otsu, 1975), are pixel selection methods. Statistical methods (e.g. spectral classification methods) that include no spatial regularization fall into this category as well. This is therefore a veritable zoo of methods we are mentioning here, and research is still active in this domain.

Of course thresholding and related methods are usually very fast and easily made interactive, which is why it is still used so much. By properly pre-processing noisy, unevenly illuminated images, or by other transforms, it is surprising how many problems can be solved by interactive or automated thresholding. However, this is of course not always the case, hence the need for more sophisticated methods.

3.2.2 Contour tracking

It was realized early that (1) on the one hand, human vision is sensitive to contours, and (2) on the other hand, there is a duality between simple closed contours and objects. A simple closed contour (or surface) is one that is closed and does not self-intersect. By the Jordan theorem, in the Euclidean space, any such contour or surface delineates a single object of finite extent. There are some classical difficulties with the Jordan theorem in the discrete setting (Najman and Talbot, 2010), Chap. 1, but they can be solved by selecting proper object/background connectivities, or by using a suitable graph, for instance the 6-connected hexagonal grid or the Khalimsky topology (Khalimsky et al., 1990; Daragon et al., 2002).

A contour can be defined locally: it is a frontier separating two objects (or an object and its background in the binary case), while an object usually cannot: an object can have an arbitrary extent. A gradient (first derivative) or a Laplacian (second derivative) operator can be used to define an object border in many cases, and gradients are less sensitive to illumination conditions than pixel values. As a result, contour detection through the use of gradient or Laplacian operators became popular, and eventually led to the Marr-Hildreth theory (1980).

Given this, it is only natural that most segmentation methods use contour information directly in some ways, and we will revisit this shortly. Early methods used *only* this information to detect contours and then tried to combine them in some ways. By far the most popular and successful version of this approach is the Canny edge detector (1986). In his classical paper, Canny proposed a closed-form optimal 1D edge detector assuming the presence of additive white Gaussian noise, and successfully proposed a 2D extension involving edge tracking using non-maxima suppression with hysteresis.

One problem with this approach is that there is no optimality condition in 2D, no topology or connectivity constraints and no way to impose markers in the final result. All we get is a series of contours, which may or may not be helpful. Finding suitable combination of detected contours (which can be incomplete) to define objects is then a combinatorial problem of high

complexity. Finally this approach extends even less to 3D.

Overall, in practical terms, these contour tracking methods are surpassed by more recent methods and should not be used without good reasons. For instance, more recent minimal-path methods can be assimilated to contour tracking methods, although they are much more sophisticated in principle (Appleton and Sun, 2003; Cohen and Kimmel, 1997). To this class of methods belongs also the "intelligent scissors"-types. There were many attempts in previous decades to provide automated delineating tools in various image processing software packages, but a good contribution was provided relatively recently by Mortensen and Barrett (1998). This method is strictly interactive, in the sense that it is designed for human interaction and feedback as opposed to being a possibility or an option. "Intelligent scissor" methods are useful to clinicians for providing ground truth data for instance. Such methods are still strictly 2D. As far as we know, no really satisfying 3D live-wire/intelligent scissor method is in broad use today(Ardon and Cohen, 2006). However, minimal surfaces methods, which we will describe shortly in section 3.4.3, in some ways do perform this extension to n-D (Grady, 2006a).

3.2.3 Statistical methods

The opposite approach to contour detection is to work on the objects, or regions themselves. An early and intuitive approach has been to try divide (the *splitting* step) an image into uniform regions, for example using a hierarchical representation of an image in the form of quadtrees (in 2D) and octrees (in 3D). Uniformity can be defined by statistical parameters and/or tests. Subsequently, a *merging* step considering neighboring and statistical region information is performed (Horowitz and Pavlidis, 1974). Initial considered statistics were color and intensity, but other region descriptors can be used as well, for instance including texture, motion and so on. In this approach, even though regions statistics are used, they are inevitably derived at the pixel level. The split and merge approach consists of acquiring all the statistics first and to base a decision on them.

A different, also productive approach consists of building a *model* first. One way is to consider an image as a 2D or 3D graph of pixels, to start from a vast over-segmentation at the pixel level, and to evolve cliques of pixels (e.g. sets of one, two or more pixels that are fully-connected, respectively called unary, binary or higher-level cliques) to fit that model. This is the *Markov Random Field* (MRF) model, named in this way by comparison to classical one-dimensional Markov chains, for which only immediate neighboring relationships matter. Models that can be written using these cliques turn out to correspond to energies featuring weighted finite sums with as many terms as there are different kinds of cliques. (Geman and Geman, 1984) proposed to optimize these sums using Gibbs sampling (a form of Monte-Carlo Markov Chain algorithm) and simulated annealing. This was first used for image restoration, but can be readily applied to segmentation as well. This approach was very successful because it is very flexible. Markers, texture terms can all be added in, and many algorithmic improvement were proposed over the years. However, it remains a relatively costly and slow approach. Even though Geman and Geman showed that their simulated annealing strategy can converge under some circumstances, it only does so under conditions that make the algorithm extremely slow, and so usually only a non-converged or approximate result is used. More recently, it was realized that Graph-Cut methods were well-suited to optimized some MRF energies very efficiently. We will give more details in the corresponding section.

MRFs belong to the larger class of Bayesian methods. Information-theoretic perspectives and formulations, such as following the Minimum Description Length principle also exist.

These frameworks are also very flexible, allowing for example region competition (Zhu and Yuille, 2002). However the corresponding models might be complicated both to understand and run, and sometimes possess many parameters that are not obvious to tune. Well-designed methods are guaranteed to converge to at least a local minimum.

In general, when dealing with regions that have complex content (for instance textures, or multispectral content), statistical methods can be a very good choice although they cannot be recommended for general work, since simpler and faster methods often are sufficient.

3.2.4 Continuous optimization methods

In the late 1980s, it was realized that contour tracking methods were too limited for practical use. Indeed, getting closed contours around objects was difficult to obtain with contour tracking. This meant that detecting actual objects was difficult except in the simplest cases.

3.2.4.1 Active contours

Researchers therefore proposed to start from already-closed loops, and to make them evolve in such a way that they would converge towards the true contours of the image. Thus were introduced *active contours*, or *snakes* (Kass et al., 1988). The formulation of snakes takes the following continuous-domain shape:

$$E_{\text{snake}} = \int_0^1 E_{\text{internal}}(\mathbf{v}(s)) + E_{\text{data}}(\mathbf{v}(s)) + E_{\text{constraints}}(\mathbf{v}(s))ds \tag{3.1}$$

Without entering into too much detail, this model is very flexible. It contains internal terms, image data terms and constraints terms:

- The first term, the internal energy, contains a curvature term and a "rubber band" energy. The former tends to smooth the resulting contour following a thin plate, while the latter tends to make it shrink around features of interest. Other terms such as kinetic energy can be added too, which makes it possible for the snake to avoid some noisy zones or go past some flat areas.
- The second term, the data energy, attracts the active contours towards points of interests in the image: typically image contours (zones of high gradient), lines or termination points.
- The last term, the constraint term, is optional, but allows interaction with the snake by defining zones of attraction and repulsion.

To solve this equation, the Euler-Lagrange of (3.1) is worked out (typically in closed form), and a gradient descent algorithm is used. All the terms are combined in a linear combination fashion, allowing them to be balanced according to the needs of the user. Due to this flexibility, the active contour model was very popular in the literature as well as in applications. It fits very well into the interactive segmentation paradigm because constraints can be added very easily, and it can be quite fast, because it uses a so-called Lagrangian framework: the contour itself is discretized at regular interval points and evolves according to (3.1). Convergence towards a local minimum of the energy is guaranteed, but may require many iterations.

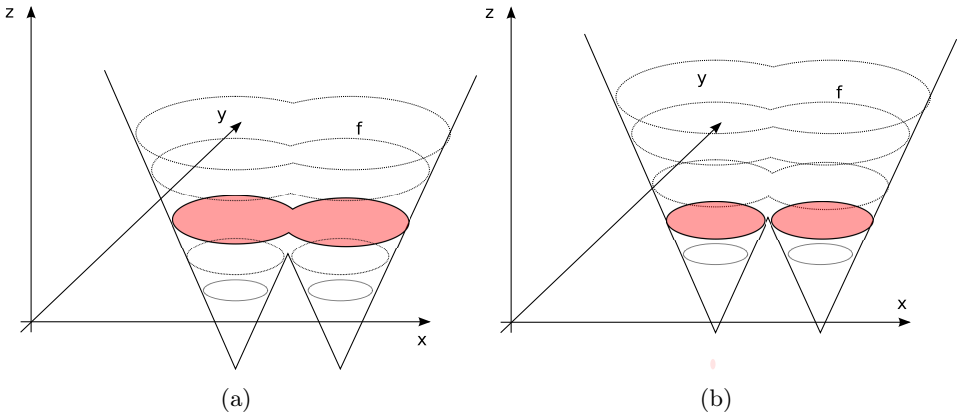


Figure 3.1: Embedding and evolving a curve as a level set of a higher-dimension function. The zero-level of function ψ is shown in color, representing a 2D contour. To evolve the contour, the whole function evolves. Note that topology changes can occur in the contour, while the embedding surface shows no such effect.

In practice, there are some difficulties: the snake energy is flexible but difficult to tune. Because of the contour evolution, points along the contour tend to spread out or bunch up, requiring regular and frequent resampling. There can also be topological difficulties, for instance causing the snake to self-intersect. The snake is also sensitive to its parametrization and to initialization. Finally, even though a local optimum is guaranteed, in practice it may not be of good quality due to noise sensitivity.

One major difficulty with snakes is that they can be extended to 3D via triangulation, but such extensions can be complicated, and that topological problems plaguing snakes in 2D are usually more difficult to avoid in 3D. However 3D active surfaces are still widely used, because they make it easy to improve or regularize a triangulated surface obtained by other means. For instance, the brain segmentation software FreeSurfer includes such a method. To distinguish them from other models we are going to introduce now, snake-like active contours or surfaces that are sometimes called *parametric deformable models*.

3.2.4.2 Level sets

One way to avoid altogether some of the problems brought about by the way parametric deformable models are discretized, is to embed the contour into a higher-dimensional manifold. This idea gave rise to *level sets*, proposed by Osher and Sethian (1988). Remarkably, this is around the same time when active contours were proposed. However level sets were initially proposed for computational fluid dynamics and numerical simulations. They were applied to imaging somewhat later (Malladi et al., 1995; Sethian, 1999b). A contour is represented on the surface S of an evolving regular function ψ by its zero level-set, which is simply the threshold of the function ψ at zero. By using sufficiently regular embedding functions ψ, namely signed distance transforms from an initial contour, it was possible to propose effective evolution equations to solve similar problems to Lagrangian active contours.

The main advantages of the level-sets method were that contour resampling was no longer necessary, and that contour self-intersection (shock solutions) were also avoided. Because level sets were able to change topology easily (see Fig. 3.1(b)). This means practically that it was

possible, at least in theory, to initialize a segmentation by drawing a box around a series of object of interest, and that the level set could find a contour around each of them. This was seen as a major benefit by the vision community. The level set Eulerian formulation (where the whole space is discretized) is thought to offer better theoretical guarantees than the Lagrangian framework of previous non-embedded formulations, and the simulation of function evolution is a well-researched topic, with many usable and interesting results. Finally, the formulation is dimension independent: level sets work virtually unchanged in 3D or higher dimensions, which is a major benefit.

There are many drawbacks as well. First the level set formulation is more expensive than earlier active contour formulations. It requires the iterative solving of PDEs in the whole space, which is expensive. In practice it is possible to limit the computation in a narrow band around the contour, but this is still more than if they were limited to the contour itself, and requires the resampling that was sought to be avoided. The surface S of function ψ is implicitly represented by the function itself, but it requires more space than the contour. In 3D or more this could be sometimes prohibitive. Some contour motions are not representable (e.g. contour rotation), but this is a minor problem. More importantly, the fact that level-sets can undergo topology changes is actually a problem in image analysis, where it is useful to know that a contour initialized somewhere will converge to a single simple closed contour. In some cases, a contour can split or even disappear completely, leading to undesirable results.

Nonetheless, level-set formulations are even more flexible than active contours, and very complex energies solving equally complex problems have been proposed in the literature. Solving problem involving texture, motion, competing surfaces and so on is relatively easy to formulate in this context (Paragios and Deriche, 2002a,b). For this reason they were and remain popular. Complex level-set formulation tend to be sensitive to noise and can converge to a poor locally optimal solution. On the other hand, more robust, closer to convex solutions can now be solved via other means. An example of relatively simple PDE that can be solved by level sets is the following:

$$\psi_t + F|\nabla\psi_t| = 0, \tag{3.2}$$

where F is the so-called speed function. Malladi and Sethian proposed the following for F:

$$F = \frac{1 - \varepsilon\kappa}{1 + |\nabla I|} + \beta(\nabla\psi.\nabla|\nabla I|) \tag{3.3}$$

The first part of the equation is a term driving the embedding function ψ towards contours of the image with some regularity and smoothing controlled by the curvature κ. The amount of smoothing is controlled by the parameter ε. The second term is a "balloon" force that tend to expand the contour. It is expected that the contour initially be placed inside the object of interest, and that this balloon force should be reduced or eliminated after some iterations, controlled by the parameter β. We see here that even though this model is relatively simple for a level-set one, it already has a few parameters that are not obvious to set or optimize.

3.2.4.3 Geodesic active contours

An interesting attempt to solve some of the problems posed by overly general level sets was to go back and simplify the problem, arguing for consistency and a geometric interpretation of

the obtained contour. The result was the geodesic active contour, proposed by Caselles et al. (1997). The level set formulation is the following;

$$\psi_t = |\nabla\psi| \operatorname{div}\left(g(I)\frac{\nabla\psi}{|\nabla\psi|}\right) \tag{3.4}$$

This equation is virtually parameter-free, only a g function is required. This function is a *metric* and has a simple interpretation: it defines at point x the cost of a contour going through x. This metric is expected to be positive definite, and in most cases is set to be a scalar functional with values in $\mathbb{R}^+$. In other words, the GAC equation finds the solution of:

$$\operatorname{argmin}_C \int_C g(s)ds, \tag{3.5}$$

where C is a closed contour or surface. This is the minimal closed path or minimal closed surface problem, i.e. finding the closed contour (or surface) with minimum weight defined by g. In addition to simplified understanding and improved consistency, (3.4) has the required form for Weickert's PDE operator splitting (Weickert et al., 2002; Goldenberg et al., 2001) allowed PDEs to be solved using separated semi-implicit schemes for improved efficiency. These advances made GAC a reference method for segmentation, which is now widely used and implemented in many software packages such as ITK. The GAC is an important interactive segmentation method due to the importance of initial contour placement, as with all level-sets methods. Constraints such as forbidden or attracting zones can all be set trough the control of function g, which has an easy interpretation.

As an example, to attract the GAC towards zones of actual image contours, we could set

$$g \equiv \frac{1}{1+|\nabla I|^p} \tag{3.6}$$

With $p = 1$ or 2. We see that for this function, g is small (costs little) for zones where the gradient is high. Many other functions, monotonically decreasing for increasing values for ∇I, can be used instead. One point to note is that GAC have a so-called *shrinking bias*, due to the fact that the globally optimal solution for (3.5) is simply the null contour (the energy is then zero). In practice, this can be avoided with balloon forces but the model is again non-geometric. Because GAC can only find a local optimum, this is not a strong problem, but this does mean that contours are biased towards smaller solutions.

3.2.5 Graph-based methods

The solution to (3.5) proposed in the previous section was in fact inspired by preexisting discrete solution to the same problem. On computers, talking about continuous-form solutions is a bit of a misnomer. Only the mathematical formulation is continuous, the computations and the algorithms are all necessarily discrete to be computable. The idea behind discrete algorithm is to embrace this constraint and embed the discrete nature of numerical images in the formulation itself.

3.2.5.1 Graph cuts

We consider an image as a graph $\Gamma(\mathcal{V}, \mathcal{E})$ composed of n vertices $\mathcal{V}$ and m edges $\mathcal{E}$. For instance, a 2D $nx \times n_x$ 4-connected square grid image will have $n = n_x^2$ vertices and $m = 2 \times n_x \times (n_x - 1)$ edges[1]. We assume that both the edges and the vertices are weighted. The vertices will typically hold image pixel values and the edges values related to the gradient between their corresponding adjacent pixels, but this is not necessary. We assume furthermore that a segmentation of the graph can be represented as a graph *partition*, i.e:

$$V = \bigcup_{V_i \in \Gamma} V_i; \forall i \neq j, V_j \cap V_i = \emptyset. \tag{3.7}$$

Then $E^\star$ is the set of edges such that their corresponding vertices are in different partitions.

$$E^\star = \{e = \{p_i, p_j\} \in E, p_i \in V_i; p_j \in V_j, i \neq j\}. \tag{3.8}$$

The set $E^\star$ is called the *cut*, and the cost of the cut is the sum of the edge weights that belong to the cut:

$$C(E^\star) = \sum_{e \in E^\star} w_e, \tag{3.9}$$

where w_e is the weight of individual edge e. We assume these weights to be positive. Reinterpreting these weights as *capacities*, and specifying a set of vertices as connected to a *source* s and a distinct set connected to a *sink* t, the celebrated result of Ford and Fulkerson (1962) is the following:

Theorem 3.1 *Let P be a path in Γ from s to t. A* flow *through that path is a quantity which is constrained by the minimum capacity along the path. The edges with this capacity are said to be saturated, i.e. the flow that goes through them is equal to their capacity. For a finite graph, there exists a maximum flow that can go through the whole graph Γ. This maximum flow saturates a set of edges E^s. This set of edges define a cut between s and t, and this cut has minimal weight.*

This theorem is illustrated in Fig. 3.2.

In 2D and if Γ is planar, this duality essentially says that the Ford and Fulkerson minimum cut can be interpreted as a shortest path in a suitable dual graph to Γ (Appleton, 2004). In arbitrary dimension, the maxflow - mincut duality allows us to compute discrete minimal hypersurfaces by optimizing a discrete version of (3.4).

There exist many algorithms that can be used to compute the maximum flow in a graph (also called network in this framework), but none with a linear complexity. Recent augmenting paths algorithms (Boykov and Kolmogorov, 2004) are effective in 2D where the number of vertices is relatively high compared to the number of edges. In 3D and above, where the reverse is true, push-relabel algorithms Goldberg and Tarjan (1988) are often more efficient. These algorithms can only be used when there is one source and one sink. The case where there

[1]This particular computation is left as an exercise to the reader...

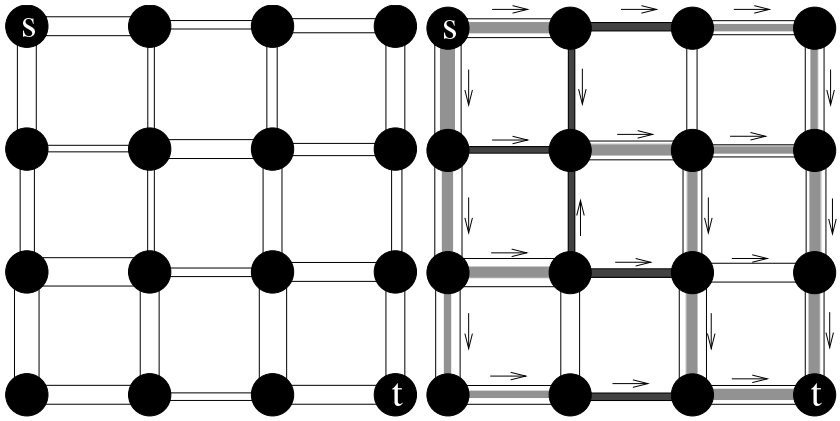


Figure 3.2: (a) A graph with edge weights interpreted as capacities, shown as varying diameters in this case. (b) A maximum flow on this graph. We see that the saturated vertices (in black) separate s from t, and they form a cut of minimum weight.

are multiple sources or sinks is known to be NP-hard. To compute energies comprising several sources or sinks and leading to multi-label segmentation, approximations can be used, such as α-expansions. These can be used to formulate and optimize complex discrete energies with MRF interpretations (Veksler, 1999; Boykov et al., 2001), but the solution is only approximate. Under some conditions, the results are not necessarily a local minimum of the energy, but can be guaranteed not to be too far from the globally optimal energy (within a known factor, often 2).

In the last ten years, Graph-Cut (GC) methods have become extremely popular due to their ability to solve a large number of problems in computer vision, particularly in stereo-vision and image restoration. In image analysis, their ability to form a globally optimal binary partition with a geometric interpretation is very useful. However, GC do have some drawbacks. They are not easy to parallelize, they are not very efficient in 3D, they have a so-called *shrinking bias*, just as GAC and continuous maxflow have as well. In addition they have a *grid bias*, meaning that they tend to find contours and surfaces that follow the principal directions of the underlying graph. This results in "blocky" artifacts, which may or may not be problematic.

Due to their relationship with sources and sinks, which can be seen as internal and external markers, as well as their ability to modify the weights in the graph to select or exclude zones, GC are at least as interactive as the continuous methods of previous sections. Much more details about this can be found in the next chapter, chapter chap:discrete-maxflow

3.2.5.2 Random walkers

In order to correct some of the problems inherent to graph cuts, Grady introduced the Random Walker (RW) in 2004 (Grady and Funka-Lea, 2004; Grady, 2005). We set ourselves in the same framework as in the Graph Cuts case with a weighted graph, but we consider from the start a multilabel problem, and, without loss of generality, we assume that the edge weights are all normalized between 0 and 1. This way, they represent the probability that a random

particle may cross a particular edge to move from a vertice to a neighboring one. Given a set of starting points on this graph for each label, the algorithm considers the probability for a particle moving freely and randomly on this weighted graph to reach any arbitrary unlabelled vertex in the graph before any other coming from the other labels. A vector of probabilities, one for each label, is therefore computed at each unlabelled vertex. The algorithm considers the computed probabilities at each vertex and assigns the label of the highest probability to that vertex.

Intuitively, if close to a label starting point the edge weights are close to 1, then its corresponding "random walker" will indeed walk around freely, and the probability to encounter it will be high. So the label is likely to spread unless some other labels are nearby. Conversely, if somewhere edge weights are low, then the RW will have trouble crossing these edges. To relate these observations to segmentation, let us assume that edge weights are high within objects and low near edge boundaries. Furthermore, suppose that a label starting point is set within an object of interest while some other labels are set outside of it. In this situation, the RW is likely to assign the same label to the entire object and no further, because it spreads quickly within the object but is essentially stopped a the boundary. Conversely the RW spreads the other labels outside the object, which are also stopped at the boundary. Eventually the whole image is labeled with the object of interest consistently labeled with a single value.

This process is similar in some way to classical segmentation procedures like seeded region growing (Adams and Bischof, 1994), but has some interesting differentiating properties and characteristics. First, even though the RW explanation sounds stochastic, in reality the probability computations are deterministic. Indeed, there is a deep relation between random walks on discrete graphs and various physical interpretations. For instance, if we equate an edge weight with an electrical resistance with the same value, thereby forming a resistance lattice, and if we set a starting label at 1 volt and all the other labels to zero volt, then the probability of the RW to reach a particular vertex will be the same as its voltage calculated by the classical Kirchhoff's laws on the resistance lattice (Doyle and Snell, 1984). The problem of computing these voltages or probability is also the same as solving the discrete Dirichlet problem for the Laplace equation, i.e. the equivalent of solving $\nabla^2\varphi = 0$ in the continuous domain with some suitable boundary conditions (Kakutani, 1945). To solve the discrete version of this equation, discrete calculus can be used (Grady and Polimeni, 2010), which in this case boils down to inverting the graph Laplacian matrix. This is not too costly as it is large but very sparse. Typically calculating the RW is less costly and more easily parallelizable than GC, as it exploits the many advances realized in numerical analysis and linear algebra or the past few decades.

The RW method has some interesting properties with respect to segmentation. It is quite robust to noise and can cope well with weak boundaries (see Fig 3.3). Remarkably, in spite of the RW being a purely discrete process, it exhibits no grid bias. This is due to the fact that level lines of the resistance distance (i.e. the resistance between a fixed node and all the others) in an infinite graph with constant edge weights are asymptotically isotropic (Cserti, 2000). RW exhibit a shrinking bias but not as strong as GC.

3.2.5.3 Watershed

While there are many variations on discrete segmentation methods, we will consider one last method: the Watershed Transform (WT). It was introduced in 1979 by Beucher and

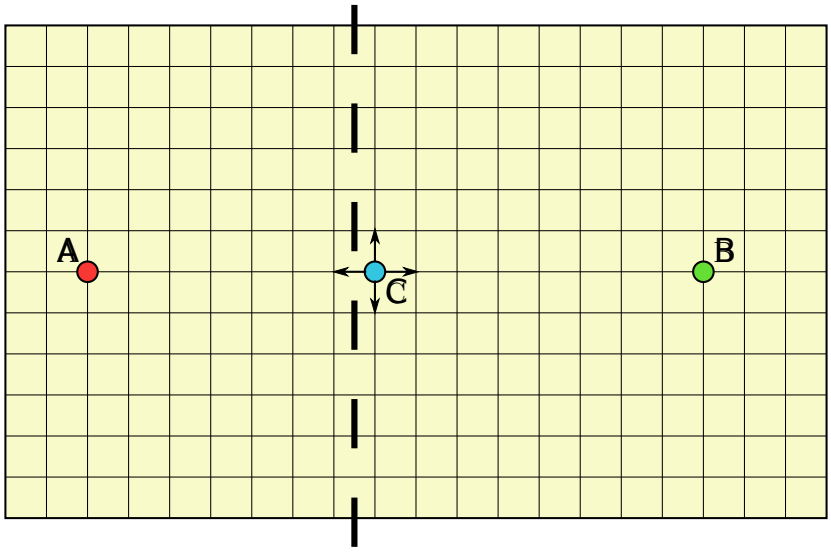


Figure 3.3: An intuitive explanation of why the Random Walker copes well with weak boundaries. We assume constant, high probabilities everywhere on this graph, except where thick vertical lines cross an edge, where the probabilities are low. A and B represent labels, and we estimate the probability of a random walker in C to move to the left as opposed to all the other directions (north, south or east). We see that locally the probabilities are identical, but globally, there are many ways for a random walker to come from B to the north, east or south position from C. However, there is only one way to move to the west of C, and that is to go through C. Therefore, Random walker probabilities must be high up to C, and then drop precipitously. Since the situation is symmetrical with respect to A, it is likely that the region left of the thick lines will be labelled with A, and the region right to it is going to be labelled with B. This is in spite of the fact that the boundary defined by the thick vertical lines is weak and closer to A than B.

Lantuéjoul (1979) by analogy to the topography feature in geography. It can be explained intuitively in the following manner: assimilate a grey-level image to a 3D topographical surface or terrain. A drop of water falling onto this surface would follow a descending path towards a local minimum of the terrain. The set of points, such that drops falling onto them would flow into the same minimum, is called a *catchment basin*. The set of points that separate catchment basins form the *watershed line*. Finally, the transform that takes an image as input and produces its set of watershed lines is called the *Watershed Transform*. To use this transform in practical segmentation settings, we must reverse the point of view somewhat. Assume now that labels are represented by lakes on this terrain and that by some flooding process, the water level rises evenly. The set of points that are such that waters from different lakes meet is also called the watershed line. Now this watershed line is more constrained, because there are only as many lines as necessary to separate all the lakes.

This intuitive presentation is useful but does not explain why the WT is useful for segmentation. As the "terrain", it is useful to consider the magnitude of the gradient of the image. On this gradient image, objects interior will have values close to zero and will be surrounded by zones of high values: the contours of the objects. They can therefore be assimilated to catchment basins, and the WT can delineate them well (see 3.4.

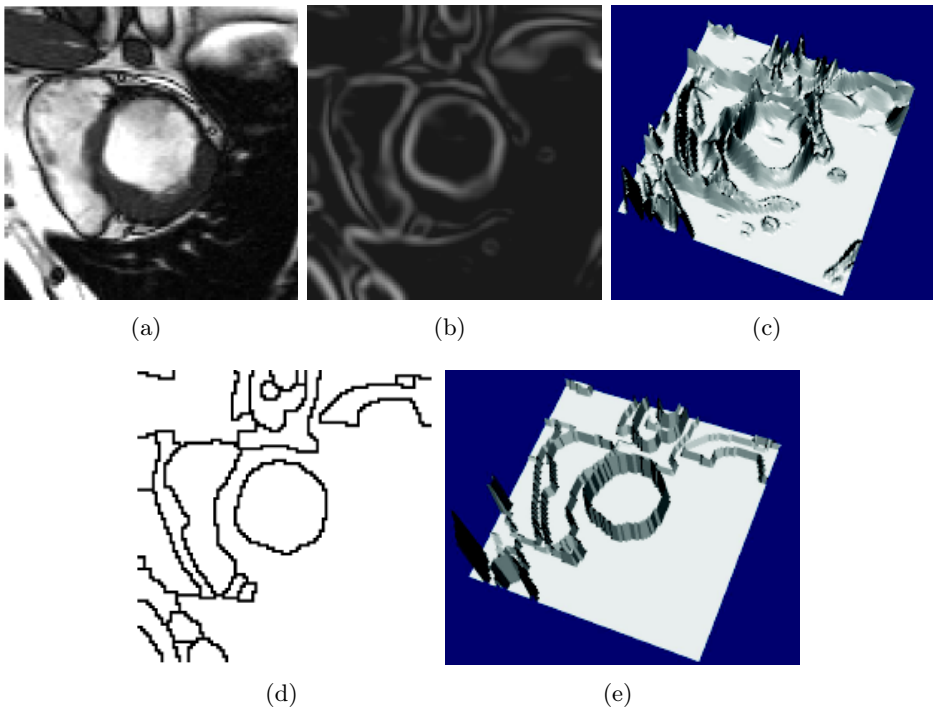


Figure 3.4: Watershed segmentation: (a) an MRI image of the heart, (b) its smoothed gradient, (c) the gradient seen as a topographic surface, (d) Watershed of the gradient, (e) topographical view of the watershed.

The WT is a seeded segmentation method, and has many interesting interpretations. If we consider the image again as a graph as in the GC setting, then on this graph the set of watershed lines from the WT forms a graph cut. The edges of this tree can be weighted with a functional derived from a gradient exactly as in the GC case. Computing the WT can be performed in several efficient ways that are broadly similar (Vincent and Soille, 1991; Meyer, 1994), but an interesting one is to consider the Maximum Spanning Forest algorithm (Cousty et al., 2008). In this algorithm, the classical graph algorithm for maximum spanning tree (MST) is run on the graph of the image, following for instance the algorithms of Kruskal (1956); Prim (1957), with the following difference: when an edge selected by the MST algorithm is connected with a seed, then all vertices that are connected with it become also labelled with this seed, and so on recursively. However, when an edge selected by the MST algorithm would be connecting two different seeds, the connection is simply not performed. It is easy to show that (1) eventually all edges of the graph are labelled with this algorithm; (2) that the set of edge that are left connected form a graph cut separating all the seeds; and (3) that the labels are connected to the seeds by subtrees. The result is a maximum spanning forest, and the set of unconnected edges form a watershed line. The maximum spanning forest algorithm can be run in quasi-linear time (Cousty et al., 2010).

3.2.6 Generic Models for Segmentation

Even though seeded models are the focus of this chapter, we say here a few words about generic models that are not seeded by default, because they contain powerful ideas for the

future of seeded models.

3.2.6.1 Continuous models

Over the years, several now widely cited formulations of the segmentation problem have been proposed, including for instance the functional of Mumford and Shah (1989) or the active contour without edges (AWE) of Chan and Vese (2001). They generally seek to solve the segmentation problem in the vision setting, and can be used for image restoration as well (denoising, inpainting, etc).

In particular, the Mumford-Shah functional is the following:

$$E(\mathbf{f}, C) = \beta \int_{\Omega} (\mathbf{f} - \mathbf{g})^2 dA + \alpha \int_{\Omega \setminus C} |\nabla \mathbf{f}|^2 dA + \gamma \int_{C} ds. \tag{3.10}$$

This formulation is very interesting because it has been an inspiration to many. In this expression, $\mathbf{g}$ is the original image, $\mathbf{f}$ a piecewise smooth approximation of $\mathbf{g}$ and C a collection of contours where $\mathbf{f}$ is discontinuous. In essence, C represents the segmentation of $\eth$ and $\mathbf{f}$ is a restored (denoised, etc) model of $\eth$. The first term in (3.10) is a data fidelity term ; the second is a total variation term (TV), and the last optimizes an unweighted contour length.

Both MS and AWE initially were solved using level-sets methods, but more recently convex methods have been used. The MS functional is NP-hard in general, but convex relaxations are computable, and can be exact in the binary case. In particular, the Rudin-Osher-Fatemi (ROF) p model is convex, and correspond to the MS model without the last term (Chan and Bresson, 2010). From the image analysis point of view, these models are not readily usable, because they correspond to simplistic models of vision, and if markers or shape constraints are added, they tend to dominate the model, which then does not help very much.

3.2.6.2 Hierarchical models

Hierarchies of segmentations are a powerful way to deal with the multi-resolution inherent to nature. Many images contain objects at different scales, in medical imaging a vascular network is a typical example. It is very difficult to come up with a seeded strategy to solve this case. One general idea is to perform many segmentations at once or in sequence, taking into account various scales. This is not as easy to do as it sounds, because simply repeating a segmentation procedure with different parameters will not yield compatible segmentations, in the sense that contours are not likely to remain stable as the scale increases or decreases. One way of dealing with this is to offer a measure of the strength of a particular piece of contour, and as the scale increases, remove pieces of contours with weak strength first. This *saliency* idea was proposed by Najman and Schmitt (2002) in the context of watershed segmentation, but more work has been done on this idea since, for example on ultrametric watershed and connections (Najman, 2011; Soille, 2008). A saliency map or ultrametric watershed is an interactive segmentation because edge strength can be selected by interactive thresholding for instance, but it is not always obvious how to combine this with seeded segmentation.

Hierarchical models do offer some other benefits, such as the ability to efficiently optimize Mumford-Shah like functional on a saliency map (Guigues et al., 2006). Other functional are also possible, such as optimizing minimum ratio costs (Grady and Schwartz, 2006). There are some drawbacks as well, such as decreased speed, and extra memory requirement, and again

the question of compatibility with other constraints. This is at present a very interesting area of research.

3.2.6.3 Convex and convex relaxation methods

Many optimization-based methods provide an exact solution only in the binary case. This is true for graph-cut methods with convex or Potts priors. This is also true for TV-based methods such as the Chan-Vese model (Nikolova et al., 2006). However for more than two labels, the problem is usually non convex (with exceptions, e.g. Random Walker, Random Forests and the Watershed-based transforms). The area of convex relaxation techniques that allow searching for high-quality approximations is currently very active (Chan and Bresson, 2010; Pock et al., 2009). In the realm of unseeded multi-label segmentation methods, a recent survey was recently published by Nieuwenhuis et al. (2013) comparing most relaxation methods for the Potts model regularization, and concluding that most methods provided comparable results but significantly varying performances. They concluded the method by Zach et al. (2008) was the fastest.

3.2.6.4 Combinations

Many segmentation algorithms can be combined to provide different sets of compromises or extensions. For instance, Yuille proposed an interesting model combining Bayesian methods with level-sets (Zhu and Yuille, 2002). An active area of research today are so-called *turbopixels*, where a first-level over-segmentation is performed in order to group pixels into consistent region of similar size. Then these regions are linked in a graph and a discrete segmentation is performed over these (Levinshtein et al., 2009). This two-level segmentation procedure has some advantages in terms of speed and resource allocations. Final segmentations can still be precise if the first-order grouping is done well, and these methods are compatible with seeded segmentation. However, segmentation quality may be poor in the presence of weak edges (Stawiaski et al., 2007).

3.3 A unifying framework for discrete seeded segmentation

In many early segmentation methods, the focus was on the values of the pixels themselves, or, in graph terms, the values of the vertices. Since the advent of graph-cut methods, it was realized that focusing instead on the edges was useful. In particular, defining a gradient function on the edges is easy. Let p and q be two vertices in the graph $\Gamma(\mathcal{V}, \mathcal{E})$ of image I, that we have been using so far (see section 3.2.5), then we can set as weight $w_{p,q}$ for the edge linking p and q any value depending on the discrete gradient $I_q - I_p$, where I_q represents the value of I at vertex q. For instance, we can use $w_{p,q} = \exp(-\beta|I_q - I_p|^2)$, with β a positive scalar parameter. This is a monotonically decreasing function of the gradient, recommended by several authors. In addition, there are topological advantages, as a cut in such a graph obeys the Jordan property in arbitrary dimension. In addition, there is a fundamental difference between regions, formed of uniformly labeled vertices, and cuts formed of edges. In former pixel-based segmentation procedures, the contours were themselves made of pixels, which created problems (Cousty et al., 2008). The only significant drawback is that storing edge weights rather than pixels costs roughly twice as much memory in 2D, or three times as much

in 3D for the simplest nearest-neighbour connectivity. This extra cost increases with the connectivity, and may indeed be a problem in some applications.

3.3.1 Discrete optimization

Assuming then this simple model of discrete images, the segmentation problem can be viewed as an optimization problem over cliques of one or two pixels, like in the MRF setting. For instance, classical graph cut can optimize the following problem exactly:

$$\text{argmin}_x E(x) = \sum_{u \in \mathcal{V}} w_u |x_u - y_u| + \sum_{(u,v) \in \mathcal{E}} w_{u,v} |x_u - x_v|, \tag{3.11}$$

in the case where x is a binary vertex labeling, y a reference binary image that can for instance represent seeds, and w_u and $w_{u,v}$ positive unary weights and binary weights respectively. The seeded segmentation case corresponds to an image y containing some vertices labelled with 0, others with 1 (the seeds) and unlabelled ones as well. The w_u for the labelled vertices in y have infinite weights, and the unlabeled one zero. Using the same notation, the Random Walker optimizes the following energy:

$$\text{argmin})_x E(x) = \sum_{u \in \mathcal{V}} w_u (x_u - y_u)^2 + \sum_{(u,v) \in \mathcal{E}} w_{u,v} (x_u - x_v)^2. \tag{3.12}$$

In this case, the optimal labelling $x^\star$ is not binary even if y is binary. It expresses the probability of a vertex to belong to label 0 or label 1. To reach a unique solution, we must threshold the result:

$$s_u = 0 \text{ if } x_u < \frac{1}{2}, s_u = 1 \text{ otherwise.} \tag{3.13}$$

In this case the binary result s represents the segmentation. There is a striking similarity between (3.11) and (3.12), which leads us to propose a unifying framework.

3.3.2 A unifying framework

We propose to optimize the following general discrete energy:

$$\text{argmin}_x E(x) = \sum_{u \in \mathcal{V}} w_u^p |x_u - y_u|^q + \sum_{(u,v) \in \mathcal{E}} w_{u,v}^p |x_u - x_v|^q, \tag{3.14}$$

The p and q terms are integer exponents. In cases where the optimal $x^\star$ is not binary, we threshold it in the end as in (3.13). An analysis of the influence of p and q provides us with Table 3.1:

In this table, we find some well-known algorithms, such as previously mentioned GR and RW, in addition to the Shortest Path Forests (Cousty et al., 2010), that uses forests of shortest path leading to seeds as segmentation criteria. Most of the other cases are not interesting (Voronoi diagrams for instance), but the case $q = 1$ or 2 and $p \to \infty$ are novel and interesting: this is the Power Watershed algorithm (Couprie et al., 2009).

q \ p	0	finite	∞
1	Collapse to seeds	Graph cuts	Power watershed $q = 1$
2	ℓ_2 norm Voronoi	Random walker	Power watershed $q = 2$
∞	ℓ_1 norm Voronoi	ℓ_1 norm Voronoi	Shortest Path Forest

Table 3.1: Our generalized scheme for image segmentation includes several popular segmentation algorithms as special cases of the parameters p and q. The power watershed are previously unknown in the literature, but may be optimized efficiently with a maximum spanning forest calculation.

3.3.3 Power watershed

Among the drawbacks of traditional watershed as described in section 3.2.5.3 are the following: (1) watershed has no energy interpretation and is purely a segmentation algorithm ; (2) watershed segmentations are not unique: for the same seed placement and edge weights, the same definition can provide different results ; (3) watershed results tend to leak in the presence of weak boundaries. We intend to solve all three problems.

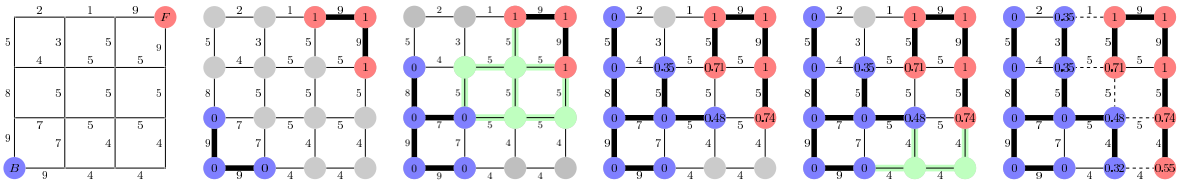
An analysis of the convergence of (3.14) in the case $q = 1$ or 2 and $p \to \infty$ led us to the following algorithm

Algorithm 1: power watershed algorithm, optimizing $p \to \infty, q \geq 1$

Data: A weighted graph $\Gamma(\mathcal{V}, \mathcal{E})$ and a reference image y containing seed information
Result: A **potential function** x and a labeling s associating a label to each vertex.
Set x values as unknown except seed values.
Sort the edges of E by decreasing order of weight.
while *any node has an unknown potential* **do**
  Find an edge (or a plateau) E_{MAX} in E of maximal weight; denote by S the set of nodes connected by E_{MAX}.
  if *S contains any nodes with known potential* **then**
    Find x_S minimizing (3.14) (using the input value of q) on the subset S with the weights in E_{MAX} set to $w_{ij} = 1$, all other weights set to $w_{ij} = 0$ and the known values of x within S fixed to their known values. Consider all x_S values produced by this operation as known.
  else
    Merge all of the nodes in S into a single node, such that when the value of x for this merged node becomes known, all merged nodes are assigned the same value of x and considered known.
Set $s_i = 1$ if $x_i \geq \frac{1}{2}$ and $s_i = 0$ otherwise.

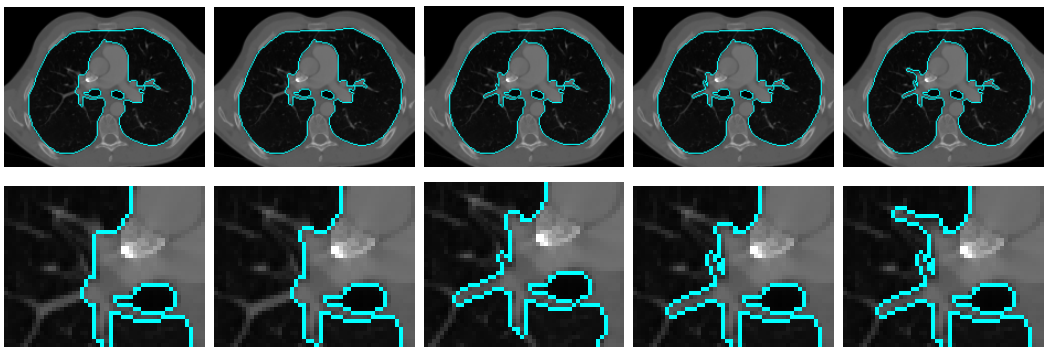
This algorithm is illustrated in Fig 3.5. We also show some pictorial results in Fig. 3.6, where we compare qualitatively the results of PW with the other classical discrete segmentation algorithms, namely GC, RW, SPF and the classical WT in the form of a Maximum Spanning Forest (MSF).

More details on the Power Watershed algorithm can be found in Couprie et al. (2011b).



(a) (b) (c) (d) (e) (f)

Figure 3.5: Illustration of the different steps for Alg 1 in the case $q = 2$. The values on the nodes correspond to x, their color to s. The bold edges represents edges belonging to a Maximum Spanning Forest . (a) A weighted graph with two seeds, all maxima of the weight function are seeded, (b) First step, the edges of maximum weight are added to the forest, (c) After several steps, the next largest edge set belongs to a plateau connected to two labeled trees, (d) Minimize (3.14) *on the subset (considering the merged nodes as a unique node) with $q = 2$ (i.e., solution of the Random Walker problem), (e) Another plateau connected to three labeled vertices is encountered, and (f) Final solutions x and s obtained after few more steps. The q-cut, which is also an MSF cut, is represented in dashed lines.*



(a) GC (b) RW (c) SPF (d) MSF (e) PW

Figure 3.6: Slides of a 3D lung segmentation. The foreground seed used for this image is a small rectangle in one slice of each lung, and the background seed is the frame of the image.

We show that the PW algorithm performs very well in terms of quantitative results, that qualitatively PW is devoid of size bias and grid artifacts, while being only slightly slower than standard watershed and much faster than either GC or RW, particularly in 3D. The PW algorithm provides a unique unambiguous result, and an energy interpretation for watershed, which allows it to be used in wider contexts as a solver, for instance in filtering (Couprie et al., 2010) and surface reconstruction. One chief advantage of PW, with respect to GC for instance, is its ability to compute a globally optimal result in the presence of multiple labels. When segmenting multiple objects this can be important.

3.4 Globally optimum continuous segmentation methods

Here we provide some arguments for globally optimal segmentation in the context of continuous-domain optimization.

3.4.1 Dealing with noise and artifacts

Even assuming we can construct a contents metric as explained in the first section, there are several sources of artifacts in segmentation: (1) weak edges cause uncertainty in the result ; (2) noise tends to corrupt boundaries, and for some methods tend to lead to wrong results ; (3) method artifacts, such as a size bias or blockiness artifacts can cause undesirable results. Of course all these artifacts are linked and essentially due to the contents metric, reflecting insufficient knowledge about the content, but this is precisely to solve this problem that we require segmentation.

Weak edges are a fact of life in medical imaging. Most often in CT for example it is difficult to delineate a lesion because it has a similar radiation absorption profile as surrounding tissues. In this case it is useful to use methods that interpolate contours and surfaces well. The Geodesic Active Contour (GAC) is very useful in this context because of its geometric formulation and shortest path/minimal surface interpretation. In addition it is straightforward to add simple shape information, such as elliptical or spherical shape priors.

Many iterative methods do not cope well with noise. One reason might be that the formulation of the corresponding energy is not convex, which implies that it would probably not have a single global optimum. This is unfortunately the case of most active contours and level set formulations, including the classic formulation of Geodesic Active Contours. In addition, these methods make it easy to add terms to the energy and make it look like it can be optimized. The reality is that in most cases, these methods get stuck into a poor quality local minimum. If models are complex, tweaking their parameters is difficult and error-prone. This is the reason why most recent segmentation models feature few parameters and tend to propose formulations that can be optimized globally.

Finally all segmentation methods present artifacts. Graph Cuts for instance tend to both produce blocky results (grid bias) and favour small objects (shrinking bias). They can be coped with by augmenting the connectivity of the graph and by metric manipulation knowing the position of the seeds. However it is better to prefer formulations that are isotropic in nature, such as continuous-domain ones.

These are some of the reasons that motivate us to mention continuous, isotropic, efficient formulations for finding the global solution to the GAC equation exactly.

3.4.2 Globally optimal Geodesic Active Contour

In spite of advances in GAC optimization, more efficient ways of solving equation (3.5) do exist. In particular, in 2D, this equation can be solved by a continuous-domain, non point-convex circular shortest path (Appleton and Sun, 2003). The solution, called the globally optimal geodesic active contour (GOGAC) is globally optimal and extremely efficient (Appleton and Talbot, 2005b), although it can only find a single contour at a time. The GOGAC solution is as flexible as the original GAC, but due to its formulation and algorithm, it is significantly less affected by noise.

This GOGAC has no shrinking bias and no grid bias, however it tends to favour circular boundaries due to its polar coordinate equivalence. This may be desirable in some applications, but not in others. It can be avoided by using a different weighting than the $1/r$ given in the original article. A flat weighting can be used if small solutions are forbidden for instance.

3.4.3 Maximal continuous flows and Total Variation

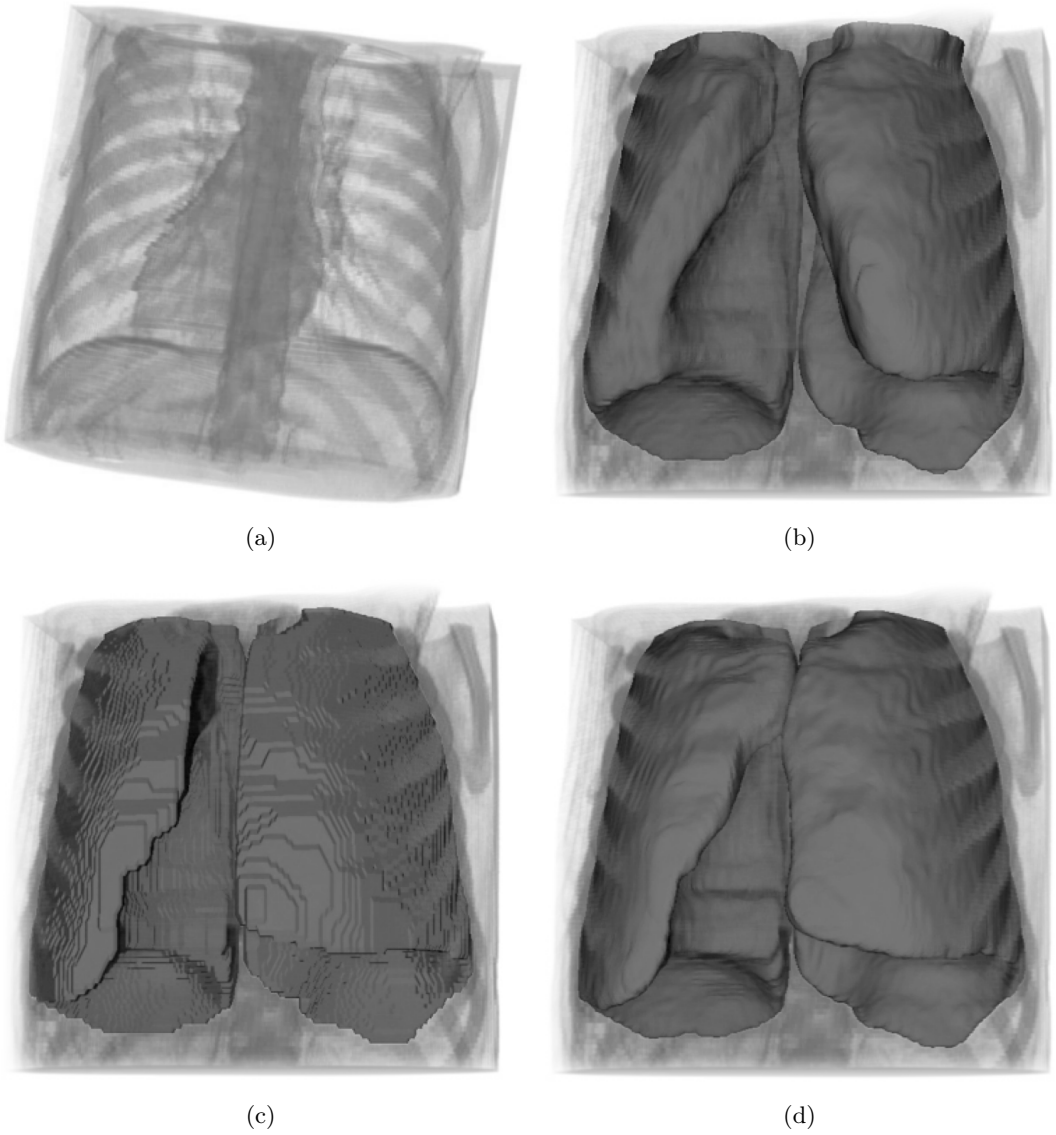
The GOGAC solution is extremely efficient but does not extend to 3D and more, but in 2006, Appleton and Talbot proposed a continuous maximum flow solution to solve this problem. Their solution, inspired by local solvers for discrete graph cuts, consists of simulating a flow originating from a source s and ending in a sink t, and a pressure field, linked by a PDE system forming a propagation equation and constrained by the metric g:

$$\begin{array}{rl} \frac{\partial \vec{F}}{\partial t} & = -\nabla P \\ \frac{\partial P}{\partial t} & = -\operatorname{div} \vec{F} \\ \|\vec{F}\| & \leq g \end{array} \tag{3.15}$$

This unusual system, at convergence, produces a scalar field P that acts as an indicator function for the interior of the contour C of (3.5). It solves the closed minimal surface problem exactly and efficiently, and so this represents a better way to solve it than (3.4). The result in 2D is exactly the same as that obtained with GOGAC. This Continuous Maximum Flow (CMF) result provides a direct algorithm for solving the problem posed by independently Iri and Strang (Iri, 1979; Strang, 1983). Interestingly, researchers in image restoration have proposed over the years solution to Strang's dual problem, that of minimizing the *total variation* (TV) of a functional. Initial solutions used level-set formulations (Rudin et al., 1992), and later ones convex optimization methods (Chambolle, 2004; Pock et al., 2009). Nonetheless, it is thought that primal maximum flow methods may be better suited to segmentation than TV formulations (Couprie et al., 2011c). Note that CMF are also biased towards small contours, and because they find a global optimum, this is a more serious problem than with standard GAC. However there exist ways to remove this shrinking bias for an arbitrary collection of sources and sinks (Appleton, 2004), and the bias is less strong in 3D and can be ignored, as long as small solutions are forbidden, using for instance large enough inner seeds. CMFs are about as fast as GC, but can be parallelized easily.

3.5 Comparison and discussion

In the space of a single chapter it is not possible to present a thorough, quantitative assessment of most popular segmentation methods. However, in table 3.2, we present a qualitative



(a) (b)

(c) (d)

Figure 3.7: Segmentation of the lungs in a chest CT image. (a) The CT image. (b) Segmentation using 3D standard Geodesic Active Contours. The surfaces fail to fill the base of the lung. (c) Segmentation using a discrete maximal flow algorithm. Observe the directional bias due to the grid. (d) Segmentation from identical input using continuous maximal flows.

comparison.

In this table, we have presented all the methods families discussed in the chapter. The figure 1 indicates a low, undesirable score and the highest figure is 5. These score are potentially controversial and represent experience and opinion rather than hard fact. We have ranked all methods according to some desirable features. In the following discussion, we present robustness as the ability of a method to cope with noise and weak edges. Flexibility denotes the ability of a method to be used in different contexts: seeded or non-seeded segmentation, the possibility to optimize different models, for instance with texture, the ability to use the

Table 3.2: A qualitative assessment of many popular segmentation methods. See text for details.

Method	High Speed	Low memory	Multi-label	Flexibility	Robustness	No bias	3D and more	Parallelizable	Multi-resolution	score
Pixel selection	5	5	1	1	1	5	5	5	1	29
Contour tracking	5	5	1	4	3	4	1	1	2	26
Split and merge	4	4	5	2	2	3	4	3	5	32
MRF - SA	1	3	4	4	3	3	3	5	2	28
Active contours	4	4	1	5	2	2	2	2	2	24
Level sets	1	2	2	5	2	3	5	4	3	27
GAC	2	2	2	3	3	3	5	4	3	27
Graph cuts	2	3	2	4	5	2	4	2	3	27
Watershed	4	4	5	2	3	5	5	3	4	35
Random Walker	3	3	5	3	4	4	4	4	3	33
GOGAC	5	3	1	1	5	3	1	1	2	22
CMF	3	2	1	2	5	3	5	4	3	28
Power watershed	4	3	5	3	4	5	5	2	4	35

method in different contexts.

Taking methods in order, we see that (1) pixel selection uses low resources but is extremely simplistic ; (2) has some speed and flexibility advantages but is limited to 2D ; (3) Split-and-merge methods generally have high scores but are not robust and not flexible; (4) MRFs and Bayesian methods optimized by simulated annealing feature a lot of flexibility but are very slow ; (5) Active contours are fast and flexible but not robust, they find only one object at a time, and cannot be extended easily to 3D ; (6) Level sets (LS) are similar in some ways but are quite slow, require lots of resources and are not robust. They do extend to 3D readily ; (7) GAC are a particular case of LS methods, which are popular in medical imaging because they are faster and more robust but less flexible. Still standard GAC is slow compared to many other methods and still not robust enough. (8) Graph cuts are a very popular recent method, which feature relatively high score across the board, in particular they are among the most flexible and robust methods. However, they are slow, particularly in 3D, not parallelizable easily and feature much bias. (9) Watershed is an old method but has definite advantages for segmentation: it is fast, bias-free and multi-label (it can segment many objects at once). However, it is not flexible or very robust. Watershed can be extended readily for multi-resolution, and due to its age, many parallel implementations exist, including hardware ones. (10) The Random Walker is a recent method which is similar in some ways to Watershed, but is significantly more robust. It requires more resources, however.

Among the newer methods presented in this chapter, (11) GOGAC solves GAC exactly and quickly in 2D, and so provides a quick robust solution, which is good for 2D interactive segmentation of single objects. However is is not flexible in its model. (12) CMF is probably among the most robust segmentation method in the literature for 3D segmentation, but it

segments only one object at a time, is not very flexible, and has no grid bias but does feature a shrinking bias. Finally (13) power watershed fits in between standard watershed and random walker. It is significantly more flexible and robust than standard watershed. Its speed is also comparable, but it uses more memory, and is less parallelizable.

The global score is probably even more subject to controversy than the individual ones, but it would tend to show that active contour methods should not be tried as a first choice method. For medical imaging, Random Walker and watershed-based methods are probably a good first choice, particularly for ease of use. It is comforting to realize that more modern methods suitable for 3D medical imaging (GC, RW, PW and CMF) are all very robust.

Many advantages presented in the literature, such as purported sub-pixel accuracy of segmentation, are not listed here because they are an illusion. The reported ability of some methods to control topology or on the contrary to let it allow to change is not necessarily a drawback or advantage either way, so we do not report it as well.

3.6 Conclusion and future work

In conclusion to this study, we argue that seeded or interactive segmentation is useful in medical imaging. Compared with model-based segmentation, seeded segmentation is more robust in actual image analysis applications, as opposed to computer vision. The ability to separate (1) seeds/markers ; (2) contour information and (3) contour optimization method is very useful, as these elements can play together to bring a higher likelihood of good results. From this point of view, we argue that segmentation is a process and not merely an operator.

In general, the literature focuses on contour placement optimization to the expense of the other two components, with some rare exceptions. This is unfortunate but understandable with respect to seeds/markers, because they are highly application dependent. Choice of methods for obtaining contour information is also limited, this is probably a good area for future research. One conclusion of this study is that contour placement optimization methods are important. More recent methods focus on optimization robustness, which is a very good thing. For someone not yet experienced in medical segmentation, simpler, more robust methods should be preferred over complex ones. Among those, power-watershed seems like a good candidate because of its combination of speed, relative robustness, ability to cope with multiple labels, absence of bias and availability (the code is easily available online). The random walker is also a very good solution, but is not generally and freely available.

We have not surveyed or compared methods that encompass shape constraints. We recognize that this is important in some medical segmentation methods, but this would require another study altogether.

Finally, at present there exists a kind of dichotomy between the way discrete and continuous-domain work and are presented. In the near future it is likely we will see methods unifying both aspects to great advantage.

Chapter 4

Discrete flow-based methods for image segmentation and filtering

In this chapter we review discrete flow-based methods for segmentation and filtering. Flow-based method have been used in the discrete domain for quite some time to solve minimum-cut and related optimization problems. Our approach, perhaps uncommon, is to start from the algorithms and their particularities, to arrive at formulations and applications. One reason for this is to show that the particular characteristics of the discrete flow algorithms carry over to the very end in sometimes unexpected fashions.

We review the motivation, the formulations and applications.

4.1 Discrete flows in networks

Historically, maximum-flow/minimum cut problems were studied by L. R. Ford, Jr. and D. R. Fulkerson (1956). Most detailed can be found in their 1962 book (Ford and Fulkerson, 1962).

4.1.1 Motivation

Maximum flow (maxflow) problems are common in graph theory as well as in applications. They are a form of *transport problem* common in optimization theory. The objective of the maxflow problem is to maximize the flow of a quantity through a network, when the transport cost itself is not considered.

In the process of finding this maximum throughput, Ford and Fulkerson showed that another dual quantity is minimized, corresponding to a minimum partition cost. Consequently maxflow algorithms are also widely used in partitioning problems. Image segmentation is a particular case of those. Recently, it was realized that more complex problems such as multi-label segmentation and image restoration/denoising could be non-trivially cast as partitioning problems.

4.1.2 Flows in graphs

Let $G = \{V, \vec{E}\}$ be a graph with n vertices V and m directed edges $\vec{E}$. We denote $v_i \in V$ a vertex and $e_{ij} \in \vec{E}$ the edge linking vertex v_i to vertex v_j. We also denote $\mathcal{N}(v_i)$ all the neighbours of vertex v_i, i.e. all the vertices e_j such that there exist an edge linking e_i to e_j or vice-versa.

We consider the case where the edges are weighted, and are interpreted as *capacities*. For the edge linking vertices i to j, we denote C_{ij} this capacity. Let s and t be two disjoint vertices of this graph, called respectively the source and the sink. A *flow* from s to t is a function $F : \vec{E} \longrightarrow \mathbb{R}$ associating each edge $e_{ij} \in \vec{E}$ with values in $\mathbb{R}$ with the following properties:

- Flow sign: the flow coming into a vertex, i.e. for a vertex v_i, any flow from a vertex v_j into v_i along an edge e_{ji} is counted positively. Conversely, any flow coming out of a vertex is counted negatively, i.e from vertex v_i to a vertex v_j along e_{ij}. This can be formulated in the following form:

$$\forall e_{ij} \in \vec{E}, F(e_{ij}) = -F(e_{ji}) \tag{4.1}$$

- Conservation of flow: the total (signed) flow in and out of any vertex is zero:

$$\forall v_i, \forall v_j \in \mathcal{N}(v_j), \sum_j F(e_{ij)} = 0 \tag{4.2}$$

- Capacity constraint : the flow along any edge is less or equal to its capacity:

$$\forall e_{ij}, F(e_{ij}) \leq C_{ij} \tag{4.3}$$

A graph endowed with capacities and flow is called a *network*. We will say that an edge is *saturated* if the flow going through it is equal to its capacity.

For consistency, we may add a directed edge between t and s with infinite capacity so that the total flow in the network is conserved. Our objective will shortly be to maximize the flow through the network or equivalently consequently through this extra edge. We denote the total (signed) flow through the graph as $|F|$ as the flow exiting the source s, and we write:

$$|F| = \sum_{j, e_{sj} \in \vec{E}} f_{sj} = f_{ts}$$

4.1.3 Minimum cost graph cuts

A seemingly distinct problem to the previous one is the problem of minimizing the cost of a graph cut. For this we need some new definitions.

Given a graph $G = \{V, \vec{E}\}$, let $\mathcal{P}(V)$ be a partition of the vertices of this graph. Let $\mathcal{P}(V) = \{V_1, V_2, \ldots, V_p\}$, we have

$$V = \bigcup_{i=1,\ldots,p} V_i \qquad \text{and} \qquad \forall i \neq j, V_i \cap V_i = \emptyset. \tag{4.4}$$

Note that we do not require the V_i to be connected within themselves. Given such a partition $\mathcal{P}$, its *cut* is the collection of vertices that link two different vertex subsets V_i and V_j. We will denote $\vec{E}^\star$ this collection.

$$\vec{E}^\star = \{e_{pq} \mid v_p \in V_i, v_q \in V_j, i \neq j\}. \tag{4.5}$$

The *cost* $\mathcal{K}(\vec{E}^\star)$ of this cut is the sum of the capacities C of the edges belonging to the cut.

$$\mathcal{K}(\vec{E}^\star) = \sum_{e_{pq} \in \vec{E}^\star} C_{pq} \tag{4.6}$$

We will now define an s-t cut as a cut obtained when special vertices s and t respectively lie in two different partitions. A *minimum* s-t cut is one such with minimal cost. In general, it is not unique. It is possible to generalize to arbitrary numbers of sources and sink, but in the remainder, we are only considering the *binary* case, with a single s and a single t, because it is the only case for which there exist a polynomial-complexity algorithm.

4.1.4 Images as graphs

As the topic of this chapter is image analysis, it is useful to establish notation for images as well. We assume an image can be represented as a regular graph, i.e. each pixel p is a vertex in a graph $G = \{V, \vec{E}\}$, which is connected to neighbours as in a regular grid. The dimensionality of the grid is not specified, but is usually 2D or 3D. In 2D most common grids are the familiar 4-connected or 8-connected ones, meaning that most pixels, except at the border, have respectively 4 or 8 neighbours. In 3D the 6-connected, 18-connected or 26-connected grids can be considered. We will not usually specify the dimensions of the image, but if necessary, a 2D image will be arranged in a $\{1, \ldots, n_x\} \times \{1, \ldots, n_y\}$ pixels grid. The number of pixels will then be $n = n_x.n_y$. In 3D, we will have a similar arrangement with an extra dimension with voxels arranged in n_z planes, the number of voxels will be $n = n_x.n_y.n_z$. Most often in images the weights on the m vertices are symmetric, and so the graph is undirected, but this is not the general case, so we will continue to consider an directed graph with edges in $\vec{E}$.

4.2 The Ford-Fulkerson maxflow-mincut theorem

Given the definitions above, the Ford and Fulkerson maxflow-mincut theorem is the following:

Theorem 4.1 (Ford-Fulkerson maxflow-mincut) *A maximum $\{s$-$t\}$ flow in a graph $G = \{V, \vec{E}\}$ is equal to the cost of a minimum s-t cut.*

Proof : If we write both problems as linear programs, they are dual to one another.

The primal formulation is a straightforward interpretation of the maxflow definition given in section 4.1.2. The dual is correct from the algebraic point of view, but needs more explanation. In fact every s-t cut determines a feasible solution of this dual as follows:

$$d(e_{ij}) = \begin{cases} 1 & \text{if } e_{ij} \text{ belongs to the cut} \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

$$p_i = \begin{cases} 1 & \text{if } i \text{ belongs to the partition that includes } s \\ 0 & \text{if } i \text{ belongs to the partition that includes } t \end{cases} \tag{4.8}$$

Table 4.1: Maxflow-mincut duality formulation

Maximum flow (primal)	Minimum cut (dual)
Maximize $\|F\| = \sum_{j,e_{sj}\in\vec{E}} f_{sj}$	Minimize $\sum_{e_{ij}\in\vec{E}} C_{ij}d(e_{ij})$
Subject to:	Subject to:
$\forall e_{ij} \in \vec{E}, F(e_{ij}) \leq C_{ij}$	$\forall e_{ij} \in \vec{E}, d(e_{ij}) - p_i + p_j \geq 0$
$\forall v_i \in V, \sum_j F(e_{ij}) + \sum_j F(e_{ji}) = 0$	$p_s = 1$
$\forall e_{ij} \in \vec{E}, F(e_{ij}) \geq 0$	$p_t = 0$
	$\forall i, v_i \in V, p_i \geq 0$
	$\forall e_{ij} \in \vec{E}, d(e_{ij}) \geq 0$

It is also easy to verify that the cost of this cut is the expected one, i.e. $\sum_{e_{ij}\in\vec{E}^\star} C_{ij}$.

With this, the maxflow-mincut theorem is nothing more than a particular instance of the strong linear programming duality theorem (Bradley et al., 1977, Chapter 4), which states that the optimums for the primal and dual problems are the same. □

More details about this classical result can be found in Papadimitriou and Steiglitz (1998), particularly chapter 3 and 6. From the dual formulation, an important remark is that the edges forming the cut are necessarily saturated.

4.3 Examples and remarks

Even simple examples can be enlightening in this context. On figure 4.1 we have a minimal example where the maxflow has been computed. With the extra *t*-*s* edge, all nodes are fully balanced. A minimum cut with cost 7 is represented by the saturated edges.

To say:

- saturated edges = mincut
- mincut = shortest path in dual graph in 2D, minimal surface in 3D.

Even though the maxflow-mincut problem is a linear program, it is inefficient to solve it using standard linear programming methods such as the simplex. For this reason, specific algorithms have been developed to solve it.

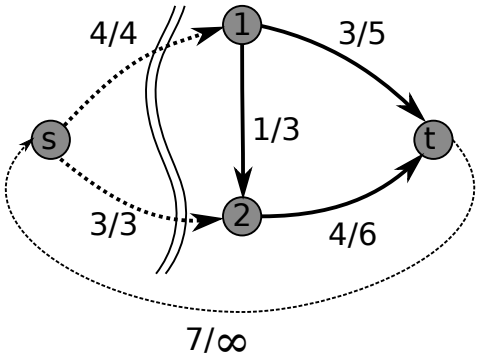


Figure 4.1: A very simple maxflow on a directed graph. The thin dashed t-s edge is optional, but it balances the whole network. The maximum flow is the total flow exiting s, which is equal to the flow through the edge $\{ts\}$. The thick dashed edges are the saturated edges separating s from t. They represent the mincut. Note that the flow through these edges is the maximum flow.

4.4 Augmenting path algorithms

In addition to proving the maximum flow minimum cut duality, Ford and Fulkerson proposed a general algorithm for the computation of a maximum flow. This algorithm successively builds a maximum flow from a source s to a sink t by repeatedly locating paths along which more flow may be pushed. At each step in this algorithm the flow is *feasible*: it satisfies the conservation constraint. Once there are no more unsaturated paths between the source and the sink, the flow is maximal. As above we consider a directed graph.

For this algorithm, we use the network $G(V, \vec{E})$ defined above, with capacities C_{ij}, on which we define a flow $F(e_{ij})$, initially set to zero. At each step, the Ford and Fulkerson algorithm produces a *feasible* flow, in the sense that it obeys equations (4.1), (4.2), and (4.3) at each step of the algorithm.

We also need to maintain a *residual network* $G^f(V, \vec{E}^f)$, with capacity $C^f_{ij} = C_{ij} - F(e_{ij})$, called the *residual capacity* and no flow. We note that it is possible that a flow from j to i be allowed in the residual network even though it may not be possible in the original one. For instance, if $F(e_{ji}) > 0$, $C_{ji} = 0$, then $C^f_{ji} = C_{ji} - F(e_{ji}) = 0 + F(e_{ij}) > 0$. In this instance, it means we can scale the flow back in the original network, which is perfectly natural.

Searching for an augmenting path in G^f can be done either breadth-first or depth-first. The breadth-first version is called the algorithm of Edmonds and Karp (1972). The original Ford-Fulkerson algorithm uses depth-first search.

The algorithm terminates when no further augmenting path can be found. At this point, s will not be able to reach t in the residual network. This means that the original network will feature a set of saturated edges, such that any path leading from s to t will pass through at least one of these edges. This mean that there exist a set of edges in the original network that separate s from t via saturated edges. This set of edges therefore must include a cut. Since this cut was formed by augmenting the flow through the original network, the flow must be maximal and the cut must be of minimal cost. The resulting flow is not necessarily unique. Once a maximum flow has been obtained, a minimum cut may then be extracted by removing the saturated edges from the graph and extracting the connected components. It is possible that more than one cut of equal, minimum value exists, in which case any one of these cuts

Algorithm 2: Augmenting paths algorithm

Data: A network $G(V, \vec{E})$ with capacities C
A network $G^f(V, \vec{E}^f)$ with capacities C^f
A source node s and a sink node t
Result: A flow $F(e_{ij})$ which is maximum for G
Set $F = 0$ on each edge of G
Set $C^f = C$ on each edge of G^f;
while *there exists a path p in* G^f *from s to t, so that* $\forall e_{ij} \in p, C^f(e_{ij}) > 0$ **do**
  find $C^f(p) = \min\{C^f(e_{ij}), e_{ij} \in p\}$
  for *all edges* $e_{ij} \in p$ **do**
    $F(e_{ij}) \longleftarrow F(e_{ij}) + C^f(p)$ (*Augment the flow uniformly along the path*)
    $C^f(e_{ij}) \longleftarrow C(e_{ij}) - F(e_{ij})$ (*Update the residual capacity along the path*)
    $C^f(e_{ji}) \longleftarrow C(e_{ji}) + F(e_{ij})$ (*Update the reverse residual capacity along the path*)

may be selected arbitrarily.

When the capacities are integers, the runtime complexity of the Ford and Fulkerson algorithm is bounded by $O(|E|L)$, where $|E|$ is the number of edges in the graph and L the number of levels (or the highest capacity). This is due to the fact that each *while* loop of the algorithm is guaranteed to raise the total flow by at least one unit. When the capacities are not rational, there are examples of graphs where the Ford and Fulkerson algorithm does not converge. However, the Edmonds-Karp algorithm is guaranteed to converge with a runtime complexity of $O(|E|^2|V|)$, where $|V|$ is the number of vertices in the graph. Note that this complexity is independent of the flow and capacity values.

Augmenting path algorithms form a large class of efficient algorithms, but there exists at least two other classes of algorithms for solving the maxflow-mincut problem.

4.5 Push-relabel algorithms

Goldberg and Tarjan proposed an interesting alternative to the augmenting path algorithm. Here we summarise the discussion presented in Sedgewick (2002) of the push-relabel algorithm, sometime called *pre-flow push* algorithm.

A *pre-flow* is a relaxed form of a flow F, which satisfies the following constraint:

- Pre-flow constraint: The inward flow to a vertex is greater than or equal to the outward flow from that vertex.

A vertex which has greater inward flow than outward flow is called an *active* vertex, with the *excess* being the positive difference between the two. The algorithm proceeds by repeatedly pushing flow outwards from active vertices toward the sink. A *height* function $H : V \to \mathbb{Z}^+$ is introduced on the vertices in order to guide the flow along the shortest unsaturated path toward the sink. The height of each vertex approximates the distance from that vertex to the sink via the unsaturated edges in the graph. The source and sink have fixed heights of $|V|$ and 0 respectively and the sink may never become active.

Here we present the algorithm of Goldberg and Tarjan. Active vertices are stored in a queue, Q. As was the case with the augmenting flow algorithm, this general pre-flow push

Algorithm 3: Push-relabel maxflow algorithm

Data: A network $G(V, \vec{E})$ with capacities C
A source node s and a sink node t
A queue Q
A height scalar $H(v_i)$ associated with each vertex
Result: A flow $F(e_{ij})$, which is maximum for G
Set $F = 0$ on each edge of G
Let H be the length of the shortest unweighted path to the sink t, and set $H(v_i) = |V|$
Set the source s as *active* and place it in Q
while *Q is not empty* **do**
  retrieve an *active* vertex v from Q
  for *all neighboring vertices u of v* **do**
    if *the edge e_{vu} is unsaturated and $H(v) = H(u) + 1$* **then**
      while *e_{vu} is not saturated and v still has excess* **do**
        Increase $F(e_{vu})$
      if *Flow to u has increased* **then**
        Set u as *active*
        Place u in Q
    if *v still has positive excess* **then**
      Increment $H(v)$
      Place v in Q
    else
      Set v as *inactive*

algorithm forms the basis of a class of maximum flow algorithms. A proof that this algorithm gives a maximum flow at termination is given in Sedgewick (2002).

Individual algorithms differ on the function and implementation of the queue Q as well as a number of additional heuristics which have been found to reduce running times in practice:

Priority queue: An effective choice of queue is a priority queue, using the height H of each active vertex as its priority. This causes the algorithm to focus its attention on pushing flow out of the vertices which are furthest from the sink, ideally examining each vertex only a small number of times. As the heights are integers we may use an efficient, integer priority queue which handles insertions, removals and priority changes in $O(1)$ time.

Global relabelling: Ideally H should resemble the length of the shortest unsaturated path to the source, so that the flows do not take unnecessarily tortuous paths. The global relabelling heuristic periodically recomputes H to ensure this, labelling vertices v which have no unsaturated path to the sink with height $\max\{H(v),\ |V| + 1\}$. To ensure that this heuristic incurs constant amortised cost the global relabelling is only performed once per $O(|V|)$ iterations.

Gap relabelling: A *gap* forms in the priority queue at height h when no active vertices have height h but there exist active vertices of both lesser height and greater height. Recall that flow can only be pushed from a vertex v to a neighbouring vertex u if $H(v) = H(u) + 1$. Note also that the height function H can only increase with each iteration. So the vertices with height greater than h cannot reach the sink, and therefore these vertices may be relabelled with height $|V| + 1$.

The algorithm may be further improved by noting that when an *active* vertex was unable to reduce its excess to 0, rather than merely incrementing $H(v)$ we may increase $H(v)$ to one greater than the maximum of its neighbours. The combination of all of these heuristics gives a practical implementation of the pre-flow push algorithm with good performance on a large class of graphs, including the regular multidimensional grids we usually encounter in image analysis.

Finally, we note that while the augmenting flows algorithm operated on the graph as a whole, the basic pre-flow push algorithm (without relabelling heuristics) operates only on a single vertex and its neighbours at a time. This property of locality will later inspire an approach to the continuous generalisation of maximum flows and consequently minimum cuts.

4.6 Classical algorithms and recent advances

Algorithms for computing the maxflow/mincut have been an important topic of research, even more so now since applications demand solving transport and flow problems on very large graphs by linear optimization standards, and since architecture changes are driving the need for parallel methods.

Table 4.2 summarizes a subset of classical general-purpose algorithms that exploit the maxflow-mincut duality.

Table 4.2: Table of general-purpose maxflow-mincut algorithms

Algorithm	Complexity	Remarks
Ford and Fulkerson (1956)	$O(V\|\text{maxlabel}\|)$	May not converge if labels are not integers.
Edmonds and Karp (1972)	$O(V.E^2)$	Variation of FF that finds paths by breadth-first search. Guaranteed to always converge.
Dinic (1970)	$O(V^2.E)$	Variant of EK with blocking flows.
General Push-Relabel (Goldberg and Tarjan, 1988)	$O(V^2.E)$	First preflow method.
P-R with FIFO vertex rule (Cormen et al., 2001)	$O(V^3)$	In many "real-world" situations, P-R algorithms variants are faster than augmenting paths.
P-R with highest active vertex rule	$O(V^2.\sqrt{E})$	
P-R with dynamic tree data structure	$O(V^2.\log(V^2/E))$	

Min-cut only algorithm and Hochbaum's conjecture: D. Hochbaum made the remark that at the end of the maxflow algorithms, the min-cut can be deduced from the max-flow, but not the other way around. She conjectures that there may exist more efficient algorithms that compute only the min-cut and dispense with the max-flow. Indeed there exist such algorithms, the most well-know of which is the Stoer-Wagner algorithm, with a runtime complexity of $O(V.E + V^2 \log(E))$, but in practice they are not the fastest methods for dealing with image-related data.

Specialized image-related algorithms Most images feature relatively few grey-levels and regular grids with symmetries that can be exploited to provide faster algorithm than general-purpose ones. The recent interest in minimum cost graph-cut algorithms in computer vision is due in no small part to such a dedicated algorithm, proposed by Boykov and Kolmogorov (2004). This algorithm uses augmenting path without a residual graph, but uses instead some well-founded heuristics. In particular, instead of searching a new path at each iteration, it remembers the paths already found and reuses them to find bottlenecks faster.

Compared with the classical algorithms, the BK algorithm can be 4-25 times faster on 2D regular networks, depending on the problem. On 3D examples, it performs about as well as PR, but get progressively worse as the graph connectivity increases. In their 2004 paper, BK do not provide a polynomial bound for their algorithm.

Other authors have proposed improvements to the BK algorithm. For instance, Juan and Boykov (2007) provide a weak polynomial bound for a related algorithm to BK and improve the 3D performance.

4.7 Applications

The message from the previous sections is that there exists fast methods to solve the maxflow/mincut optimization problem. In the vision literature, these methods are very popular and and simply called "Graph Cuts" (GC) methods. How GC relates to imaging and vision is the topic of this section. Here we briefly survey the type of applications where such methods are useful

4.7.1 Seeded image segmentation

The most natural application of GC is binary, seeded segmentation. Image segmentation is described in chapter 3, and refers to the tessellation of images into consistent regions of interests. Binary here means there are only two regions: foreground and background. Seeded means the location of both these regions is indicated by markers.

To solve this problem, we can consider the image as a graph $G(V, E)$, with each pixel considered a vertex and neighboring pixels linked with edges. The capacity (or weight) of each edge is set to a function of the gradient of the image $g(\nabla I)$. In this case, we consider the discrete gradient $\nabla I = w_{ij} = I(j) - I(i)$. Classically, one considers monotonically decreasing function of the magnitude of the gradient, such as $g \equiv \exp(-\alpha|\nabla I|^2)$ or $g \equiv 1/(1+|\nabla I|^p)$, with $p = 1$ or 2. Finally, we can connect the foreground seeds to the source s and the background seeds to the sink t. With such a setting, the maxflow/mincut problem corresponds to solving

the following problem:

$$\begin{aligned} &\underset{E^\star}{\operatorname{argmin}} \sum_{e_{ij}\in E^\star} g(|\nabla I|) \\ &\qquad s.t. \begin{cases} V = V_s \cup V_t \\ V_s \cap V_t = \emptyset \\ s \in V_s, t \in V_t \\ e_{ij} \in E^\star \iff e_i \in V_s, e_j \in V_t \end{cases} \end{aligned} \tag{4.9}$$

In other words, this finds the set $E^\star$ that tessellates the graph $G(V, E)$ into two distinct subsets, one containing s, the other t, with minimal cost. This situation and its solution is illustrated on Fig 4.2. This is the discrete equivalent to the Geodesic Active Contour (GAC) equation (3.4) of chapter 3.

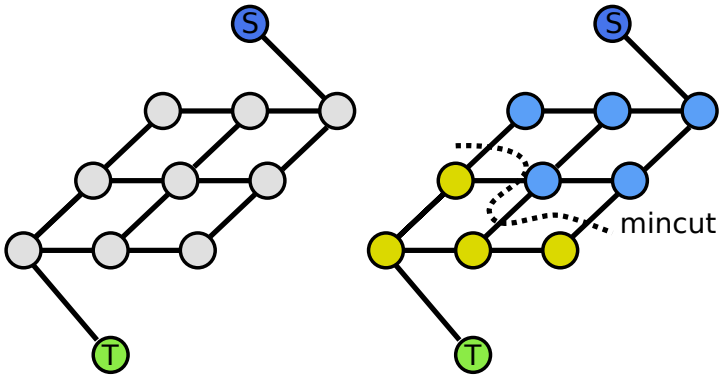


Figure 4.2: Simple binary seeded graph and associated segmentation

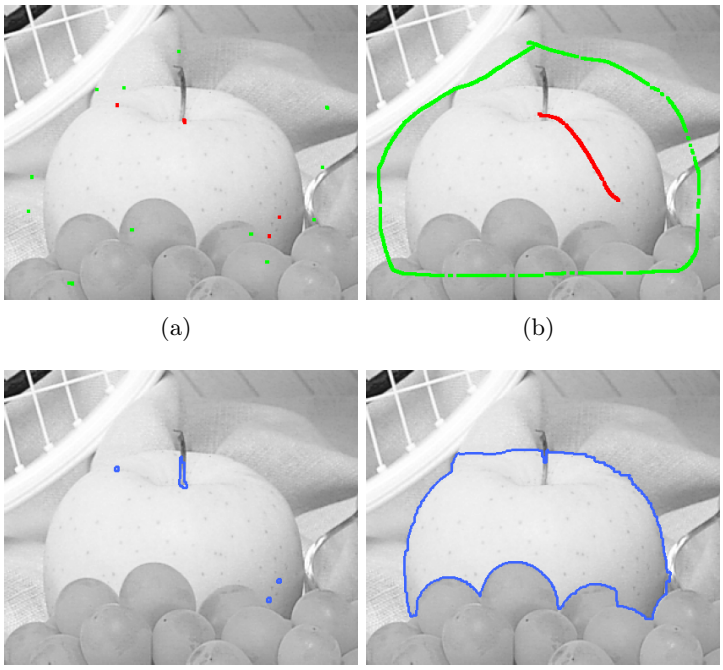
An example actual segmentation with this model is given on Fig 4.3. While simple, this example is fairly representative of GC seeded segmentations: with relatively good seed placement, the result is quite good. The method is relatively insensitive to seed placement, but tends to favour small contours, when seeds are too small. Also the final result may be look relatively blocky due to grid bias.

In order to ameliorate these problems, the binary segmentation method can be extended using so-called unary weights.

4.7.1.1 Segmentation using unary weights:

A relatively simple extension to this scheme is add so-called *unary* terms between s and t respectively to all the pixels in the image (Boykov and Funka-Lea, 2006), as on Fig 4.4. These are simply extra weighted edges drawn from both the source and the sink to every pixel in the image. The unary weights can be derived from simple intensity or colour differences. They can also be derived from local measure such as texture, or any arbitrary measure giving rise to a scalar weight. Conventionally, s can be associated to the foreground and t to the background. Topology constraints and user interaction can be enforced through the use of some strong edge weights either to the source or the sink.

The more traditional marker-controlled segmentation procedure of previous paragraphs is nothing more than a special case where only topology constraints are enforced. On Fig. 4.5,



(a) (b)

(c) (d)

Figure 4.3: Grey level segmentation of an apple. This type of CG segmentation is robust to seed placement, but tends to favour small boundaries and readily features grid bias.

we show a segmentation of red blood cells using unary weights, in this case a distance between each pixels and either pure white or pure black. In this case we did not use any hard seed to enforce topology.

An important issue is the increased graph complexity that arise when considering unary weights, that slows down computations. This has been recently studied by Lermé et al. (2010), showing that it is possible to still achieve reasonable computation speed even with these extra weights.

4.7.2 Maximum A-Posteriori - Markov Random Fields interpretation

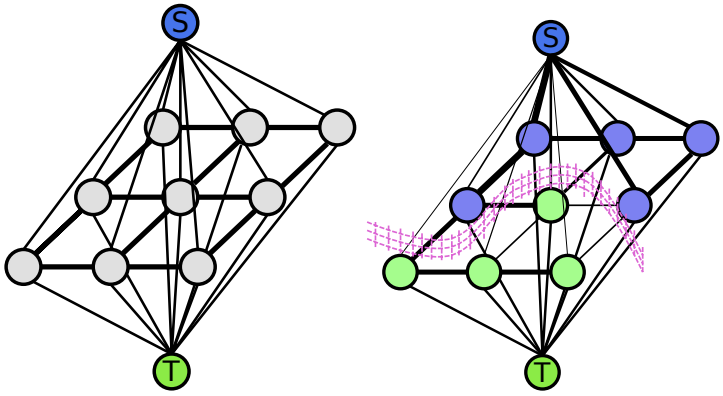
One very important aspect of graph cut optimization is their interpretation in terms of the Maximum A Posteriori - Markov Random Fields approach (MAP-MRF).

In broad terms, an MRF is a model consisting of the following items: a set of sites (a.k.a. pixels) $S = \{s_1, \ldots, s_n\}$, a set of random variable $y = \{y_1, \ldots, y_n\}$ associated with each pixel, and a set of neighbors $\mathcal{N}_{1\ldots n}$ at each pixel location. The set $\mathcal{N}_p$ describes the neighborhood of pixel p, for instance it contains the indices of the subset of S connected to p.

Furthermore, the model must obey the Markov condition, which is that

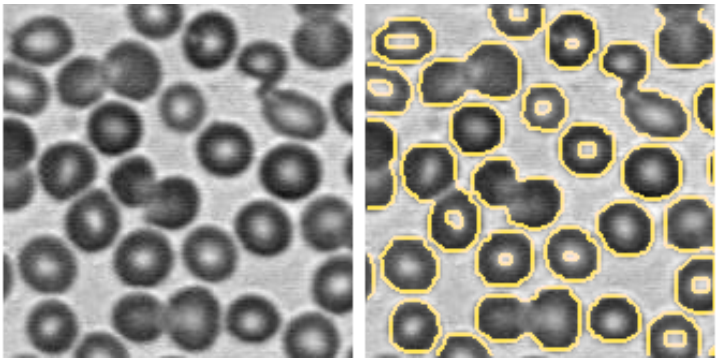
$$Pr(y_p|y_{S\setminus p}) = Pr(y_p|y_{\mathcal{N}_p}), \forall p \in S \tag{4.10}$$

In other words, the probability of y_q depends only on its immediate neighbours, not on the whole image.



(a) (b)

Figure 4.4: Segmentation with unary weights. In this case weighted edges link the source and the sink to all the pixels in the image (a). The min-cut is a surface separating s from t (b). Some strong edge weights can ensure the surface crosses the pixel plane, enforcing topology constraints.



(a) (b)

Figure 4.5: Binary segmentation with unary weights

Now, given a set of observables $\mathbf{x} = \{x_1, \ldots, x_n\}$, we can write, using Bayes' rule:

$$\hat{y} = \underset{y_{1\ldots n}}{\operatorname{argmax}}\ Pr(y_{1\ldots n}|\mathbf{x}) \tag{4.11}$$

$$= \underset{y_{1\ldots n}}{\operatorname{argmax}} \prod_{n=1}^{n} Pr(x_n|y_n) Pr(y_{1\ldots n}) \tag{4.12}$$

$$= \underset{y_{1\ldots n}}{\operatorname{argmax}} \sum_{n=1}^{n} \log[Pr(x_n|y_n)] + \log[Pr(y_{1\ldots n})] \tag{4.13}$$

$$= \underset{y_{1\ldots n}}{\operatorname{argmin}} \sum_{n=1}^{n} U_p(y_p) + \sum_{u \in \mathcal{N}_p} P_{u,p}(y_u, y_p) \tag{4.14}$$

This last sum contains unary terms $U_p(y_p)$ and pairwise terms $P_{u,p}(y_u, y_p)$. With some

restrictions, graph cuts can optimize these energies.

As an illustration, the binary segmentation problem with unary weights above can be written:

$$\operatorname{argmin} \hat{E}(G) = \sum_{v_i \in V} w_i(V_i) + \lambda \sum_{e_{ij} \in \vec{E}} w_{ij} \delta_{V_i \neq V_j} \tag{4.15}$$

In equation (4.15), V_i is 1 if $v_i \in V_s$ and 0 if $v_i \in V_t$, i.e. it is 1 if pixel i belongs to the partition containing s and 0 otherwise. $\delta_{V_i \neq V_j}$ is 1 if the corresponding e_{ij} is on the cut, and 0 otherwise. The first sum contains the pairwise terms, and sums the cost of the cut in the image plane. The second sum contains the unary terms, and adds the cost of a pixel to belong to either the partition containing s or the partition containing t.

This energy corresponds to the energy of an MRF, has a MAP interpretation, and can be optimized exactly using mincut/maxflow techniques.

4.8 A general discrete framework for energy minimization

At this stage we know how to perform binary segmentation with and without unary weights, and we have a MAP-MRF interpretation of this problem. The next question is can we extend these results to other settings? For instance can we perform image denoising or other vision-related applications within this framework?

Image restoration can indeed be written as a MAP-MRF problem, for instance consider the following formulation. Once more we consider a graph $G = \{V, \vec{E}\}$. Let $\overline{x}$ be observations of an unknown x, under the condition that x takes values over a finite set of labels $\mathcal{L}$. These can represent for instance grey-level values in an image. We can define $\mathcal{L}$ as an ordered discrete set of labels $\{1, \ldots, Q\}$. We denote x_u the label assigned to node $u \in V$. The unknown x^* is a minimum argument of the following energy function:

$$E(x) = \sum_{u_i \in V} \mathcal{D}(x_i) + \lambda \sum_{e_{i,j} \in \vec{E}} \mathcal{R}(x_i, x_j), \tag{4.16}$$

where λ is a positive real value. $D(x_i)$ is often called the *data fidelity* term and $\mathcal{R}(x_i, x_j)$ the regularization or smoothness *prior*.

A common choice of data term $\mathcal{D}$ is a pixelwise distance $\mathcal{D} = |x_i - \overline{x}_i|^p$ between the desired labeling x and a reference $\overline{x}$, representing noisy acquired data, where p is a small positive integer, e.g. 1 or 2.

Many choices of $\mathcal{R}$ lead to useful algorithms and results. A common model is the so-called Potts model, where $\mathcal{R}(x_i, x_j) = w_{ij} \min(1, |x_i - x_j|)$, and w_{ij} are spatially variant positive pairwise weights. This model corresponds to a piecewise constant prior. Other choices for $\mathcal{R}$ include $\mathcal{R} = w_{ij}|x_i - x_j|^q$, where q is typically 1 or 2 for linear and quadratic priors respectively. The latter represents an "everywhere smooth" prior with good denoising properties and lack of staircase effect in the result, but with blurred boundaries.

In recent years, energy-based optimization methods using discrete flow-based solutions, (Graph Cuts) have become very popular in computer vision applications (Boykov et al.,

1998; Ishikawa, 2003; Boykov et al., 2001). GC optimization has been applied to e.g. stereo-vision (Woodford et al., 2008), multiview reconstruction (Sinha et al., 2011), motion analysis (Xiao and Shah, 2007), segmentation (Boykov and Jolly, 2002) and image restoration (Darbon and Sigelle, 2006b). GC methods tend to provide optimal or near-optimal solutions to classical Markov Random Fields (MRF) problems, with some guarantees and in reasonable time, unlike earlier methods like Simulated Annealing (SA) (Geman and Geman, 1984) or Iterated Conditional Modes (ICM) (Besag, 1986). From the algorithmic point of view, GC problems can be solved exactly when the energy is submodular, which was shown for the binary case (binary $\mathcal{L}$) in Murota (2000); Kolmogorov and Zabih (2004) and for multi-label cases in Schlesinger and Flach (2006)

One way to solve such problems is to present the optimization problem as a segmentation problem. This was proposed by Ishikawa and Geiger (1999).

4.8.1 The Ishikawa construction

In this approach, a graph is built to represent an image represented on a finite (usually small) number Q of discrete levels. We assume the image is represented on a discrete set of vertices $\mathbb{V}$, e.g. for a regular, 4-connected, 2D image, $\mathbb{V} = \{1, \ldots, nx\} \times \{1, \ldots, ny\}$. The Ishikawa construction involves representing the quantized levels of the image in the graph itself.

The graph $G = \{V, \vec{E}\}$ is defined as follows:

1. $V = \mathbb{V} \times \{1, \ldots, Q\} \cup \{s, t\}$ is the set of vertices quantized over Q levels, together with two special vertices, the *source* s and the *sink* t.

2. $\vec{E} = \vec{E}_D \cup \vec{E}_C \cup \vec{E}_P$ is the set of edges. As before, an oriented edge from vertex a to vertex b is denoted by $e_{a,b}$. We have :

   (a) $\vec{E}_D = \bigcup_{v \in \mathbb{V}} \vec{E}_D^v$ is the upward columns of the graph. For all $v \in \mathbb{V}$, let $h_{v,k}$ denote the node in column v and row k. A single column associated with pixel v is defined as
   $$\vec{E}_D^v = \left\{e_{s,h_{v,1}}\right\} \cup \left\{e_{h_{v,k},h_{v,k+1}} \,|, k \in \{1, ..., Q-1\}\right\} \cup \{h_{v,Q}, t\}\,,$$
   (b) $\vec{E}_C = \bigcup_{v \in \mathbb{V}} \vec{E}_C^v$ the downward columns of the graph, with
   $\vec{E}_C^v = \left\{e_{h_{v,k},h_{v,k+1}} \mid k \in \{1, ..., Q-1\}\right\}$,
   (c) and the penalty edges of the graph are thus defined as
   $$\vec{E}_P = \left\{e_{h_{v,k},h_{w,k}} \mid \{v, w\} \text{ neighbours} \in \mathbb{V},\, k \in \{1, ..., Q\}\right\}.$$

The above graph is depicted in Fig. 4.6. In this figure, for simplicity we assume the image is one-dimensional, i.e. each pixel has only two neighbours, which allows us to represent the graph in a 2D planar layout. For 2D or 3D images, correspondingly more penalty edges between all neighbours in $\mathbb{V}$ are present in the graph. For such images, it is best to see the arrangement of v vertices as in the original images, with the column of penalty edges in an extra dimension.

4.8.2 Image restoration using the Ishikawa graph

Let us consider the separable function defined in (4.16) with $\mathcal{D}(x_i) = |x_i - \overline{x}_i|$, i.e. an ℓ_1 data fidelity term, and $\mathcal{R} = w_{ij}|x_i - x_j|$, i.e. also an ℓ_1 regularization term. This energy

formulation is convex and has been studied by many authors. With $w_{ij} \equiv 1$ everywhere, this regularization term is the discrete total variation (TV). Minimizing (4.16) with these two terms corresponds to a discrete formulation of the ROF model. With w_{ij} positive, this is the weighted TV formulation (Darbon and Sigelle, 2006a).

We now set the weights on the Ishikawa graph to solve this problem in the following manner:

1. The links to the source are set to an infinite capacity:

$$\forall v \in \mathbb{V}, c\left(e_{s,h_{v,1}}\right) = +\infty. \tag{4.17}$$

2. The data fidelity terms for any pixel $v \in \mathbb{V}$ is

$$\forall k \in \{1, \ldots, Q-1\}, c\left(e_{h_{v,k},h_{v,k+1}}\right) = |r_k - f(v)|, \tag{4.18}$$
$$c\left(e_{h_{v,Q},t}\right) = |r_Q - f(v)|. \tag{4.19}$$

3. The capacity of downward columns is infinite to constrain a single cut per column:

$$\forall v \in \mathbb{V}, \forall k \in \{1, \ldots, Q-1\}, c\left(e_{h_{v,k+1},h_{v,k}}\right) = +\infty. \tag{4.20}$$

4. The regularization term along the penalty edges of the graph is:

$$\text{for every } \{v, w\} \text{ neighbours } \in \mathbb{V}, \forall k \in \{1, \ldots, Q\}, c\left(e_{h_{v,k},h_{w,k}}\right) = \lambda \tag{4.21}$$

The above graph G can be extended to any convex function $\mathcal{R}$ (Ishikawa and Geiger, 1999). The capacities of $\mathcal{E}$ are adjusted in such a way that a cut of $\mathcal{G}$ corresponds to the solution of (4.16), granted by the following result:

Proposition 4.2 *With any convex function $\mathcal{R}$ and any positive function $\mathcal{D}$, then the min cut of $G = (V, \vec{E})$ as described by* (4.17) *through* (4.21) *is the globally optimal solution to* (4.16).

Proof : This result is derived from the construction of the graph. First note that we build here a binary flow network with one source and one sink. As seen in section 4.2, any binary cut that separates s and t along a series of edges, that can be interpreted as a solution $x_{\mathcal{P}}$. Indeed, the infinite capacity of the downward edges ensure a single cut edge in each column of the graph, and the infinite capacity of the upward $e_{s,h_{v,1}}$ edges for all v ensures that, in all columns, this cut will be located above one of the nodes corresponding to a level $k \in \{1, \ldots, Q\}$. We can therefore associate the cut in column v with the value of the level immediately below the cut, and associate this with $x_{\mathcal{P}}(v)$. Recalling that all labels below the cut will have the same label as s, and all that above the cut the same label as t, the value of $x_{\mathcal{P}}$ at pixel v is the highest level l in column v of the graph that is labelled like the source s. Here, by convention, the source is labelled with 1 and the sink with 0. We can then write $x_{\mathcal{P}}(v) = \max\{k, h_{v,k} = 1\}$.

Now, the computation of the maxflow/mincut on this graph minimizes the energy of the cut, interpreted as the sum of two terms:

1. since the downward constraint edges ensure a single cut edge along each column of the graph, this corresponds to contribution of the data fidelity term $\mathcal{R}$ to the total energy.
2. Similarly, we note that each penalty edges in $\vec{E}_P$ with capacity λ can be cut at most once. Let u and v be two neighbouring pixels in the graph. The cut along penalty edges between $x_{\mathcal{P}}(u)$ and $x_{\mathcal{P}}(v)$ crosses exactly as many penalty edges as there are quantization level differences between u and v. We note that this correspond to a contribution of $\lambda|x_{\mathcal{P}}(u) - x_{\mathcal{P}}(v)|$ to the total energy.

Hence, the computation of the maxflow/mincut on this graph solves (4.16) exactly. □

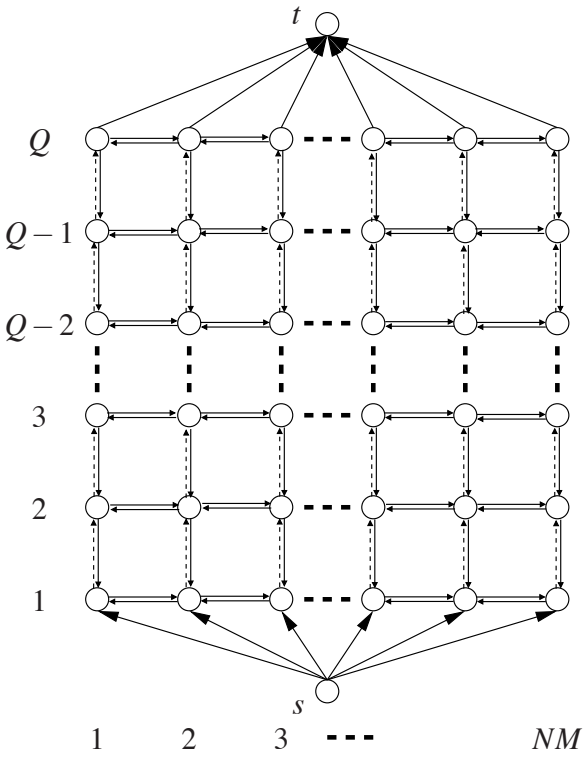


Figure 4.6: Construction of the Ishikawa optimization graph. Arrows represent the edges $\vec{E}$ and circles the nodes in V. Horizontal edges are in $\vec{E}_P$, the dotted upward vertical edges are in $\vec{E}_D$ and the plain downward vertical edges are in $\vec{E}_C$. Vertices s and t are respectively the source and the sink. All pixels in the image from 1 to n are represented in the columns. In actual 2D images, there exist many more penalty edges $\mathcal{E}_P$ than depicted here: all those between neighbours in $\mathbb{V}$.

Remark 4.3

1. *Optimizing* (4.16) *when $\mathcal{R}$ is convex and not necessarily the ℓ_1 norm is done by adding non-horizontal penalty edges (Ishikawa, 2003). These extra edges are weighted with the finite-difference approximation of the gradient of $\mathcal{R}$.*

2. *In the case when the number of quantized levels Q is small (say between 1 and 32), the Ishikawa framework is very efficient.*

3. *As the dimensionality of the problem increases, so does the number of penalty edges in the graph. The cut is always an hypersurface of codimension 1.*

4. *Ishikawa recommends solving the maxflow/mincut by using a push-relabel algorithm, which makes perfect sense as the dimensionality increases, because these algorithms have an asymptotic complexity which is less than linear in the number of edges (see section 4.5).*

5. *It is worth repeating that in this framework, everything is discrete, and that optimization is carried out completely in the discrete domain. The solution found may not exactly be equal to a convex programming solution of a continuous-version of* (4.16).

Many authors have reused and improved on the Ishikawa construction, for instance (Darbon and Sigelle, 2006b,a). The ability of the Ishikawa construction to deal with non-convex data fidelity has been used with profit in remote sensing, particularly with SAR denoising (Shabou et al., 2009), and in stereo-vision. We have also used it for joint quantization and denoising in Chaux et al. (2010); Jezierska et al. (2010).

4.9 The move framework

We have seen how a restoration problem can be mapped to a graph-cut formulation. The Ishikawa construction is remarkable, in particular because it makes few hypotheses on the data fidelity (most remarkably, convexity is not required), yet can find the global discrete optimum of this formulation. The main drawbacks of the method are the time and memory requirements, which are quickly prohibitive as soon as n, m or Q increase. To improve on this, several authors have proposed other schemes based on simpler, iterative graph constructions, called *moves*.

The so-called *move* algorithms were developed to solve multi-label problems by solving a series of binary sub-problems. Following the definition given in Veksler (2007), a move algorithm is an iterative algorithm where an estimation x_{n+1} of x is a function of x_n, an earlier estimation, through a "move" space $\mathcal{M}$, i.e. $x_{n+1} \in \mathcal{M}(x_n)$. A local minimum with respect to a set of moves is reached for x if $E(x') \geq E(x)$ for any $x' \in \mathcal{M}(x)$.

GC move algorithms have typically good theoretical guarantees for quality for certain sets of regularization terms. Classical move algorithms include expansion and swap moves Veksler (1999); Boykov et al. (2001). More recently, more sophisticated moves have been proposed, e.g. range moves and fusion moves (Veksler, 2007; Kumar and Torr, 2008; Veksler, 2009; Lempitsky et al., 2010).

All are geared towards improving the quality of the solution and the speed of the algorithm. The time complexity of move algorithms usually increases steeply with the number of labels. For example, the worst-case complexity of swap moves is quadratic in the number of labels while range-moves perform even worse. However, for problems where the number of labels is relatively low, these methods can be fast enough. Hence, move algorithms scale well with connectivity, are flexible with respect to data fidelity terms, but may not scale well with the number of labels.

4.9.1 Binary representation and sub-modularity

Now in order to be able to optimize GC move energies, we need to define their *graph-representability*, i.e. the ability to build an $\{s,t\}$-graph cut associated with these energies.

Definition 4.4 (Graph-representability) *A function E of n binary variables is called* graph-representable *if there exists a graph $G = \{V, \vec{E}\} \cup \{s,t\}$ and a subset of vertices $V_0 = \{v_1, \ldots, v_n\} \subset V$, such that for any binary vector x, the value of $E(x)$ equals that of the associated minimum s-t cut up to a constant.*

We note that for those energies that are graph-representable, maxflow-mincut algorithms as described earlier in this chapter are able to compute the global optimum of this energy in polynomial time.

We also note the following lemma:

Lemma 4.5 (Additivity) *The sum of two graph-representable energies is graph-representable.*

The proof of this lemma is simple if the graphs representing the two energies are defined on the same set of vertices. In this case, simply adding the two sets of edge weights together yields a graph that represents the sum of the two energies. A missing edge in one or the other graphs is taken as having a weight of value zero. Here, we do not need to prove this lemma in the more general case.

4.9.1.1 Sub-modularity

We now consider only binary energies with unary and pairwise terms, i.e. those that correspond to (4.16), but with x binary. Then we have the following theorem:

Theorem 4.6 (Sub-modularity) *Let E be a function of a binary vector x in the form of equation* (4.16)*, which we equivalently write here as follows:*

$$E(x_1, \ldots, x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{ij}(x_i, x_j) \tag{4.22}$$

Note that we consider ordered vertices because $E^{ij}(x_i, x_j) = E^{ji}(x_j, x_j)$, and we do not want to consider the pairwise terms twice. In this context, E is graph-representable if and only if for every pairwise terms:

$$E^{ij}(0,0) + E^{ij}(1,1) \leq E^{ij}(1,0) + E^{ij}(0,1) \tag{4.23}$$

This theorem was given in the present form in Kolmogorov and Zabih (2004), however, submodularity is the discrete equivalent of the notion of convexity. This has been known since at least the 1980s, when this notion was used in economics. A thorough treatment of discrete convexity is given in Murota (2003). We reproduce here the constructive proof of Kolmogorov and Zabih (2004) because it is very enlightening.

PROOF : We begin by representing the unary terms of (4.22). We follow the convention that cutting the s-v_i edge represents $E^i(1)$ and cutting the v_i-t edge represents $E^i(0)$. For each vertex i, there are only two cases: either $E^i(0) > E^i(1)$ or the complement $E^i(1) \geq E^i(0)$. In either cases, we draw a graph containing only positive weights, as represented on Fig. 4.7. We also take advantage of the fact that adding a constant to such graphs does not change the optimization result, so in Fig. 4.7(a) we subtracted $E^i(0)$ and in Fig. 4.7(b) we subtracted $E^i(1)$. This allows us to remove one edge of the graph, and also to cope with non-positive energies.

For the pairwise terms of (4.22), there are four cases for energy $E^{i,j}$:

$$\begin{array}{|c|c|}\hline A & B \\ \hline C & D \\ \hline\end{array} = \begin{array}{|c|c|}\hline E^{i,j}(0,0) & E^{i,j}(0,1) \\ \hline E^{i,j}(1,0) & E^{i,j}(1,1) \\ \hline\end{array} \tag{4.24}$$

We decompose these four cases in the following way:

$$\begin{array}{|c|c|}\hline A & B \\ \hline C & D \\ \hline\end{array} = A + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline C-A & C-A \\ \hline\end{array} + \begin{array}{|c|c|}\hline 0 & D-C \\ \hline 0 & D-C \\ \hline\end{array} + \begin{array}{|c|c|}\hline 0 & B+C-A-D \\ \hline 0 & 0 \\ \hline\end{array} \tag{4.25}$$

In this decomposition, A is a constant, so we do not need to represent it. The following two matrices show weights that are constant respectively with i and with j, and so can simply be added to the unary weights. This is done in a similar way as above. For the first matrix, which involves only the s-v_i and the v_i-t edges, we weight them either with $C-A$, $A-C$ or zero, in such a way that all weights are positive. The second matrix involves only the edges s-v_j and v_j-t and we do the same thing with $D-C$ or $C-D$.

The last matrix can be represented by an edge from i to j with weight $B+C-A-D$. We note that since the energy is sub-modular, (4.23) translates to $A+D \leq B+C$ and so this edge weight is always positive.

The full solution for binary weights is shown on Fig. 4.7(c). □

Here we have shown that a sub-modular energy involving unary and binary terms is always graph-representable, but the converse is also true. For simplicity we do not show the proof here, although it is not difficult.

Now we can use this general framework for applications.

4.9.2 The expansion move

As an illustration, we describe here the expansion move. This was introduced in Veksler (1999); Boykov et al. (2001). It can be use for a wide class of regularization functions $\mathcal{R}$, as long as it corresponds to a metric on the space of labels, i.e. when it obeys the following rules:

$$\mathcal{R}(x_i, x_j) \geq 0 \qquad \text{(positivity rule)} \tag{4.26}$$

$$\mathcal{R}(x_i, x_j) = 0 \Longleftrightarrow x_i = x_j \qquad \text{(equality rule)} \tag{4.27}$$

$$\mathcal{R}(x_i, x_j) = \mathcal{R}(x_j, x_i) \qquad \text{(symmetry rule)} \tag{4.28}$$

$$\mathcal{R}(x_i, x_j) + \mathcal{R}(x_j, x_k) \geq \mathcal{R}(x_i, x_k) \qquad \text{(triangular inequality)} \tag{4.29}$$

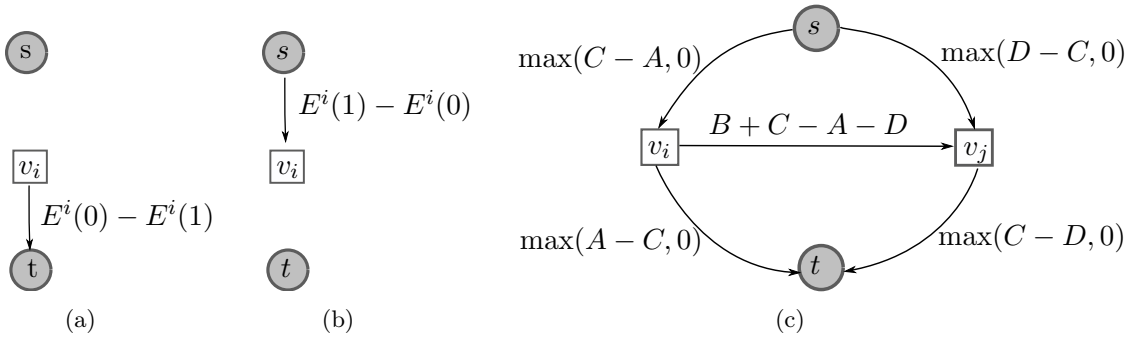


Figure 4.7: Graph-representability of unary and binary weights: (a) unary case when $E^i(0) > E^i(1)$; (b) unary case when $E^i(1) \geq E^i(0)$; (c) binary general case, where $\mathcal{R}$ is the ramp function, i.e. $\mathcal{R}(x) = x$ if $x \geq 0$ and 0 otherwise. A, B, C, D are defined in the text.

For instance, the familiar ℓ_1 distance $\mathcal{R}(x_i, x_j) = |x_i - x_j|$ obeys all of the above rules, and is therefore a metric. On the other hand, the following regularization term $\mathcal{R}(x_i, x_j) = \max(|x_i - x_j|, T)$ with $T > 0$ is not a metric. We note that we do not put restrictions on unary weights, i.e. on the data fidelity term.

Consider a particular labelling f, i.e. an arbitrary instance of $\{1, \ldots, Q\}^n$ over the n vertices of G, and a particular level $\alpha \in \{1, \ldots, Q\}$. Another labeling f' is defined to be an *expansion move* from f if for a particular vertex p, $f'(p) \neq \alpha$ implies $f'(p) = f(p)$. This means that the set of pixels assigned the label α cannot shrink when going from f to f'.

We note that it is straightforward to represent a single expansion step as a binary problem, for instance, for all $p \in V$, we can encode it with a binary vector associated with each graph vertex $x = \{x_p, p \in V\}$, such that $f'(p) = f(p) \Leftrightarrow x_p = 0$, and $f'(p) = \alpha \Leftrightarrow x_p = 1$. The following binary matrix represents the pairwise energy:

$$\begin{array}{|c|c|} \hline E^{i,j}(0,0) & E^{i,j}(0,1) \\ \hline E^{i,j}(1,0) & E^{i,j}(1,1) \\ \hline \end{array} = \begin{array}{|c|c|} \hline \mathcal{R}(x_i, x_j) & \mathcal{R}(x_i, \alpha) \\ \hline \mathcal{R}(\alpha, x_j) & \mathcal{R}(\alpha, \alpha) \\ \hline \end{array} \tag{4.30}$$

Of course the full energy associated with any expansion move includes unary terms. We include them following the convention on Fig. 4.7(a) and (b). These weights are always graph-representable.

The pairwise energy represented by (4.30) is sub-modular due to the fact that $\mathcal{R}$ is a metric. Indeed in this case $\mathcal{R}(\alpha, \alpha) = 0$, and thanks to the triangular inequality $\mathcal{R}(x_i, \alpha) + \mathcal{R}(\alpha, x_j) \geq \mathcal{R}(x_i, x_j)$, therefore (4.23) is verified.

Since an α-expansion step is sub-modular, it graph-representable. A single step of expansion can therefore be represented as in section 4.9.1. We can easily adapt Fig. 4.7 to our needs. More precisely, we consider a particular quantization level, or label α, to which we associate a graph $G_\alpha = \{\mathcal{V}, \mathcal{E}\}$, defined as follows:

1. $\mathcal{V} = V \cup \{\alpha, \overline{\alpha}\}$, where $\alpha, \overline{\alpha}$ are the two special nodes playing the role of s and t in Fig. 4.7, and V is the set of image nodes.

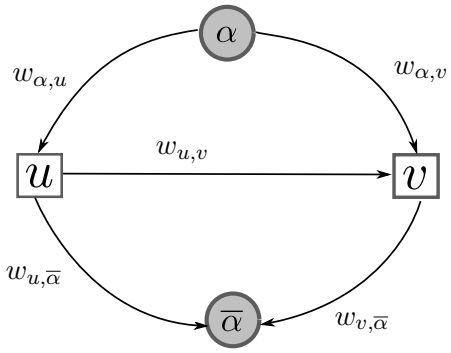


Figure 4.8: Notations for the α-expansion graph, following Kolmogorov and Zabih (2004). Here we took a simplified 2-pixel neighbourhood. The cost (or capacity) between u and v is labelled as $w_{u,v}$, and so on for all edges. The expressions for the capacity for all edges are given in the text.

2. $\mathcal{E} = \mathcal{E}_V \cup \mathcal{E}_\mathcal{N}$ is the set of edges, defined as follows :
   (a) $\mathcal{E}_V = \bigcup_{v \in V} \{e_{\alpha,v}, e_{v,\overline{\alpha}}\}$ is the set of edges between special term nodes and image nodes ;
   (b) $\mathcal{E}_\mathcal{N} = \bigcup_{\{u,v\}\text{neighbours}}$ is the set of edges between neighbours and $\mathcal{N}$ is the set of neighbours pairs containing only ordered pairs u, v, i.e. such that $u < v$.

   A simple 2-node graph with s and t is shown on Fig 4.8. The capacity for all the edges are given following (4.25), Fig. 4.7 and (4.30).

4.9.2.1 Principle and convergence of expansion moves

The principle of the expansion move algorithm is to cycle through all the labels in some order (fixed or random) and to find the lowest energy expansion move from the current labeling. If this expansion move has strictly lower energy than the current labeling, then it becomes the current labeling. Although we do not reprint a formal proof the convergence of the expansion move, we give the major arguments:

- The full energy of (4.22) is represented as a finite discrete sum, therefore it is itself always finite and discrete.
- Every expansion move strictly lowers energy (4.22) because we use this energy as reference when choosing the most effective expansion move.
- The two preceding points combined imply that expansion moves eventually converge.
- Expansion move are effective at lowering energy (4.22) because the ground state $E(0,0)$ of (4.30) is precisely the pairwise energy of (4.22), and the unary weights of any expansion move are also the unary weights of energy (4.22). This means that when no further expansion move can be effected, the resulting energy is at least a local minimum of (4.22).

The algorithm terminates with a labeling that is a local minimum of the energy with respect to expansion moves; more precisely, there is no expansion move, for any label α, with lower energy. It is also possible to prove that such a local minimum lies within a multiplicative factor of the global minimum (Boykov et al., 2001). In practice this result is seldom useful, but the bound is tight: examples can be constructed for which the expansion move will only reach this bound.

As described, the expansion move can be improved significantly from the computational perspective. In particular, although new graph weights must be recomputed at each iteration, there are ways to reuse existing flows to compute the next solution.

4.9.3 Convex and Non-convex moves

The requirement that the prior $\mathcal{R}$ be a metric is a sufficient condition for the associated expansion move to be submodular, however it is not strictly necessary. The Potts model $\mathcal{R}(x_i, x_j) = 1 - \delta(x_i - x_j)$, where δ is the Kronecker function (i.e: $\delta(0) = 1, \delta(x \neq 0) = 0$, can be approximately optimized by expansion moves. More complex moves exist, notably the $\alpha - \beta$ swap move (Veksler, 1999; Boykov et al., 2001), which is suitable for all semi-metric priors, i.e. metrics that verify (4.26) through (4.28) but not the triangular inequality (4.29). Many different moves and applications have been proposed.

We have ourselves contributed several moves and applications, which we illustrate briefly here.

4.9.3.1 Joint quantization and denoising

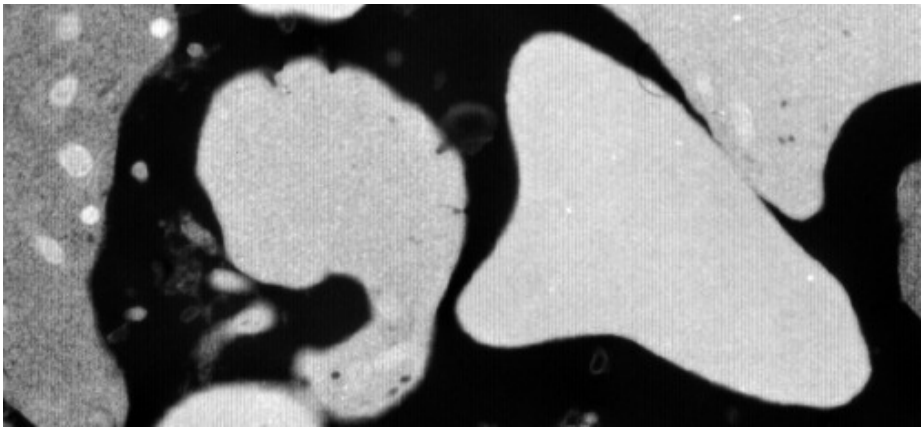
In Jezierska et al. (2010) we have proposed a joint denoising-quantization method. Quantization, defined as the act of attributing a finite number of levels to an image, is an essential task in image acquisition and coding. It is also intricately linked to various image analysis tasks, such as denoising and segmentation. We have investigated vector quantization combined with regularity constraints, a little-studied area which is of interest, in particular, when quantizing in the presence of noise or other acquisition artifacts. In this context we proposed and validated optimization methods combining continuous projection methods for quantization and graph cuts for restoration.

On Fig. 4.9 we show an example of this procedure compared with well-known Lloyd-max quantization in the case of a greyscale microscopy image.

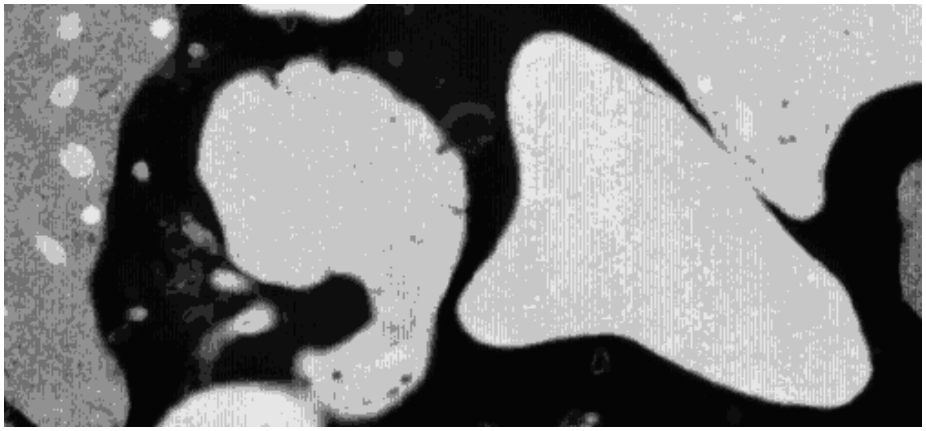
4.9.3.2 Truncated convex moves

In Jezierska et al. (2011), we proposed a joint quantizing-convex move to optimize energies with truncated convex priors. Such non-convex energies with semi-metric priors correspond to a piecewise-smooth image model. They are similar in many ways to robust metrics in model fitting and regression, particularly M-estimators with the Tukey biweight function (Huber and Ronchetti, 1981).

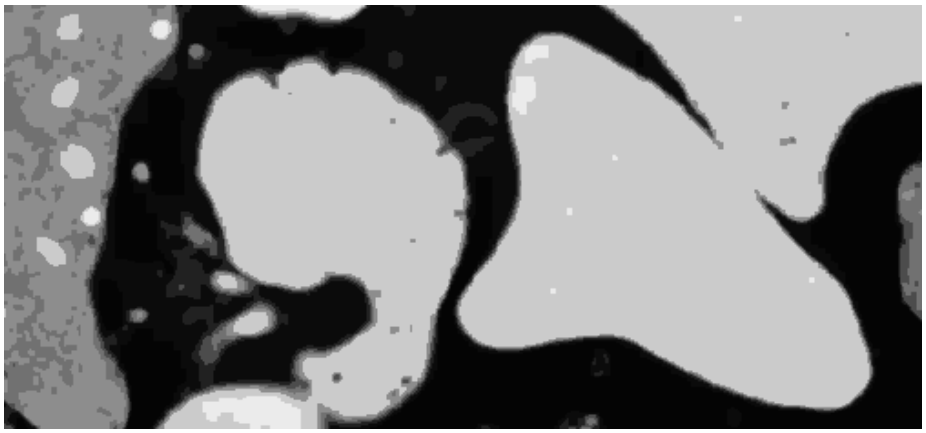
Optimizing these energies efficiently is a difficult problem. Semi-metric energies can be optimized with graph cuts using swap moves (Veksler, 1999). Expansion moves can also be used but the result is only approximate, since not all configurations are graph-representable. Belief Propagation (Felzenszwalb and Huttenlocher, 2004) and tree-reweighted methods (Kolmogorov, 2006) have also been proposed. More recently fusion moves (Lempitsky et al., 2010)



(a)



(b) LM: 0.84 bpp



(c) Ours: 0.58 bpp

Figure 4.9: Joint quantization and denoising: figures (a,b,c) illustrate a fragment of the original image, Lloyd-Max (LM) and our results, respectively. Note that LM retained the vertical acquisition artifacts as well as a lot of noise, which are absent in our result. Note also the improved entropy figure in our case.

Table 4.3: Interpretation of (4.31) *depending on the values of* p *and* q.

q \ p	0	finite	$\longrightarrow \infty$
1	Reduction to seeds	Graph cuts	Max Spanning Forest [Allène et al. 07]
2	ℓ_2-norm Voronoi	Random walker	Power watershed [Couprie et al. 09]
∞	ℓ_1-norm Voronoi	ℓ_1-norm Voronoi	Shortest Path [Sinop et al. 07]

were shown to be effective in this case. Graph-cut methods were shown to outperform BP in several cases examined by Szeliski et al. (2008)

4.10 Power watershed

It is possible and fruitful to generalize (4.16) in the following way:

$$\min_x \underbrace{\sum_{e_{ij}\in E} {w_{ij}}^p |x_i - x_j|^q}_{\text{Prior}} + \underbrace{\sum_{v_i \in V} {w_i}^p |x_i - l_i|^q}_{\text{Fidelity}} \tag{4.31}$$

For this equation to correspond to a graph, we propose the arrangement of Fig. 4.10. In these equation and figure, x is the sought after result, and l is a reference image. We note that if l is binary, then it is the same as Fig.4.4.

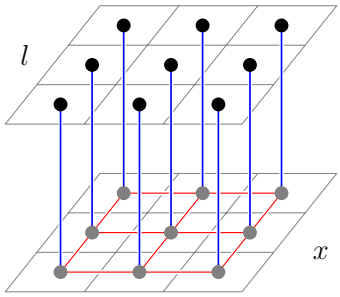
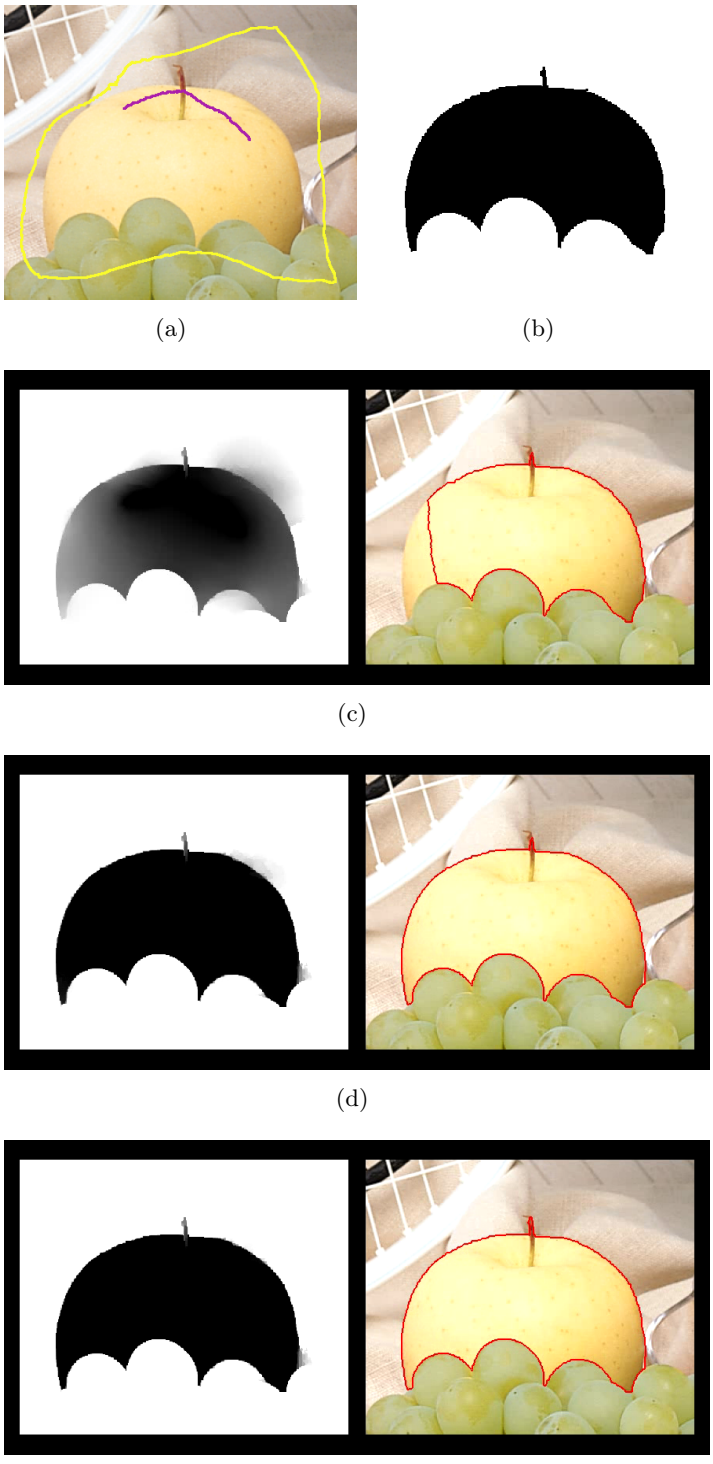


Figure 4.10: Graph arrangement for (4.31)

In this equation, we have two exponents p and q. Depending on their value, we can interpret (4.31) in several ways, indicated on Table 4.3.

Some of these interpretations are not very useful, particularly the reduction to seeds and the Voronoï results, however the relationship between graph-cuts and watershed in the case $q = 1$ was demonstrated by Allène et al. (2007) and the convergence to shortest paths forests by Sinop and Grady (2007). The case $q = 2$ and $p \longrightarrow \infty$ was novel, however. Note that the result we seek is the outcome of this convergence process (in the epi-convergence sense), and not the $p = \infty$ case, which is not interesting.



(a) (b)

(c)

(d)

(e)

Figure 4.11: Convergence of the Power Watershed. (a) Initial image with seeds; (b) Result of Power Watershed;(c) Result of Random walker with $p = 2$; (d) RW with $p = 16$: (e) RW with $p = 30$.

The convergence is illustrated on Fig. 4.11. Obviously it is not a good idea to compute the PowerWatershed in this way, so we have proposed an efficient algorithm. It boils down to the following theorems:

Theorem 4.7 (Power watershed)

- *The cut obtained by the power watershed algorithm is a MSF cut (and a watershed cut if seeds are the minima of the weights).*
- *When $q > 1$, the solution x obtained by the power watershed algorithm is unique.*
- *The power watershed result can be computed in quasi-linear time with respect to the number of pixels and in linear time with respect to the number of labels, by computing the MSF cut and a random walker on each contested plateaus.*

These and related results are all proved by Couprie et al. (2011b), reprinted as Appendix H in this document. Power watersheds share common characteristics between traditional watershed and graph-based discrete optimization methods:

- PW is quite efficient
- PW is multi-label with only a small cost associated with the number of labels. Contrary to traditional WS, this cost is non-zero however.
- PW is an optimization framework, with both unary and pairwise terms. PW can be used in similar circumstances as graph cuts or random walker beyond segmentation, e.g. for image filtering.

As illustrations we have used PW for two applications: image filtering and surface reconstruction.

4.10.1 Image filtering

Following Grady and Polimeni (2010), and denoting A the adjacency matrix of an oriented graph, we may write the anisotropic diffusion equation as

$$\frac{dx}{dt} = A^T g(Ax)Ax, \tag{4.32}$$

where x are the node weights representing the image data to be regularized. The functional g is designed to be close to zero near edges, so as to prevent edge blurring, for instance:

$$g(x) = \exp\left(-\alpha x^2\right). \tag{4.33}$$

We typically solve (4.32) with a forward Euler scheme like so:

$$x^{k+1} = x^k + dtA^T g(Ax^k)Ax^k. \tag{4.34}$$

However, Black et al. (1998) have shown that anisotropic diffusion equation (4.32) can be seen as the gradient of the energy

$$E(x) = \sigma(Ax), \tag{4.35}$$

with σ a robust estimator in the statistical sense. A gradient of the Black *et al* energy in (4.35) is given by

$$\frac{dE}{dx} = A^T \sigma'(Ax)Ax. \tag{4.36}$$

Therefore, when $\sigma'(Ax) = g(Ax)$, then the gradient of the Black et al energy, (4.36) is the same as the anisotropic diffusion equation in (4.32). In fact the $\sigma(z)$ corresponding to the Perona-Malik weighting function in (4.33) is given by the Welsch function

$$\sigma(z) = 1 - \exp\left(-\alpha z^2\right). \tag{4.37}$$

This function is a common, differentiable approximation of the ℓ_0 pseudo-norm. When used in a regularizer, as we will see in the next chapter, it is often called $\ell_2 - \ell_0$. Following classical variational formulation, we may write

$$E(x) = \sigma(Ax) + \lambda h(x, f), \tag{4.38}$$

in which f represents the intensities of the input unfiltered image, $h(x, f)$ represents a loss function (commonly the neg-log likelihood for the type of noise under study) and λ is a Lagrangian free parameter. When

$$h(x, f) = ||x - f||_2^2, \tag{4.39}$$

then the gradient of (4.38) becomes

$$\frac{dE}{dx} = A^T \sigma'(Ax)Ax + \lambda\left(f - x\right), \tag{4.40}$$

which achieves a fixed point when

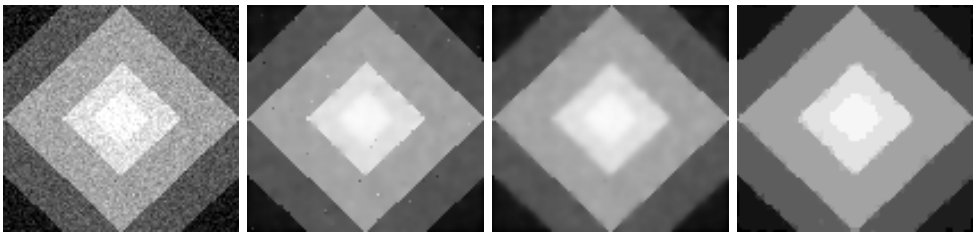
$$\left(A^T \sigma'(Ax)A + \lambda I\right) x = \lambda f. \tag{4.41}$$

Since (4.41) may be viewed as a *backward Euler* solution for the anisotropic diffusion equation (4.32) when $\lambda = \frac{1}{dt}$, then the anisotropic diffusion algorithm may be seen as the optimization of a robust estimator of the image gradient. If we alter the loss function to be the same as the regularization:

$$E(x) = \sigma(Ax) + \lambda\sigma\left(x - f\right), \tag{4.42}$$

with gradient given by

$$\frac{dE}{dx} = 2\alpha A^T \sigma'(Ax)Ax + 2\alpha\lambda\sigma'(x - f)x. \tag{4.43}$$



(a) Noisy image, SNR = 24.24dB (b) PM, SNR = 34.03dB (c) PM, SNR = 30.46dB (d) PW, SNR = 31.54dB

Figure 4.12: Comparison of Perona-Malik(PM), and power watershed(PW) algorithms for denoising a synthetic image. (b) PM used with 80 iterations $\alpha = 0.0015$, leading to a good PSNR but with remaining isolated noisy pixels. (c) PM, best compromise found for this image to remove the isolated pixels with 50 iterations and $\alpha = 0.0005$. (d) A median filtered image as initialization and $\lambda = 0.975$ allows to obtain a better SNR while removing isolated noisy pixels.

From Eq. 4.43, we can neglect the 2α. If, at any iteration, we fix the values of x inside the robust error function, then we have

$$\frac{dE}{dx^{k+1}} = A^T \sigma'(Ax^k)Ax^{k+1} + \lambda\sigma'(x^k - f)x^{k+1}. \tag{4.44}$$

This energy may be written as the steady-state optimization of the energy functional

$$\begin{aligned} E_{k+1} = & {x^{k+1}}^T A^T \sigma'(Ax^k)Ax^{k+1} \\ & -\lambda {x^{k+1}}^T \sigma'(x^k - f)x^{k+1}, \end{aligned} \tag{4.45}$$

that may also possibly be writen

$$\begin{aligned} E_{k+1} = & \sum_{e_{ij}} \sigma'(Ax^k) \left(x_i^{k+1} - x_j^{k+1}\right)^2 \\ & +\lambda \sum_{v_i} \sigma'(x^k - f) \left(x^{k+1} - f\right)^2. \end{aligned} \tag{4.46}$$

This expression for the energy to compute a minimum step is of a form that may be optimized by the *power watershed*, with $q = 2$, $p = \alpha$, the pairwise weights $w_{ij} = \exp\left(-(x_i^k - x_j^k)^2\right)^\alpha$ and the unary weights $w_i = \lambda \exp\left(-(x_i^k - y_i)^2\right)^\alpha$.

Therefore, we propose the following filtering algorithm :

1. Set $x^0 = f$.
2. Until convergence:
3. Generate weights for $\sigma'(Ax^k)$ and $\lambda\sigma'(x^k - f)$.
4. Use power watershed to optimize (4.46) to obtain x^{k+1}.

Since both the regularizer and the fidelity loss function are non-convex, the initialization is important. In the case of denoising, we achieved good result with starting from a median-filtered version of the noisy image, as shown on Fig. 4.12.

Although interesting, since it correspond to a very efficient non-convex solver, this approach needs more work. The non-convex fidelity term in particular cannot easily be interpreted in a MAP framework. More results and details can be found in Couprie et al. (2010).

4.10.2 Surface reconstruction from isolated points

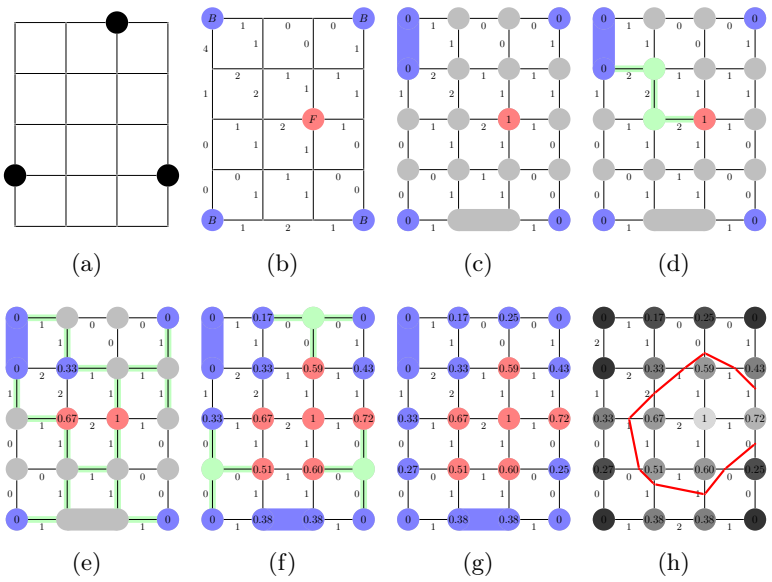


Figure 4.13: (a) Three dots in a 4×5 lattice. (b) Associated lattice weighted by an Euclidean distance map from the dots, with Foreground and Background seeds. (c) First steps of the power watershed algorithm optimizing (4.31) *in the case $q = 2$ and $p \to \infty$. Nodes with a maximum weight are merged. (d) A plateau of weight 2 (in green) including different seeded nodes is encountered. The Random walker algorithm is applied to label the nodes on the plateau. (e,f) New plateaus of weight 1 and 0 are encountered, the Random walker algorithm is applied, (g) Final labeling x solution of* (4.31)*. The resulting isocontour is represented in red.*

This problem of reconstructing a surface from isolated points, e.g. from laser range data, has been studied multiple times. In particular, recently TV and graph-cut approaches have been proposed. In this problem, we assume a discretized volume, given isolated points, as well as inner and outer markers (i.e. points that lie respectively inside the surface to be reconstructed or outside). We do not assume a particular topology for the surface, but it has to separate the inner markers from the outer markers. A graphical explanation for the algorithm in 2D is given in Fig 4.13

From then on, extension to 3D is straightforward, and some comparative results are shown on Fig.4.14. More details, results and comparisons are given in Couprie et al. (2011a). Overall, we argue that among dense reconstruction algorithms (those needing a full volume reconstruction), the Power-watershed based method offer an interesting compromise with respect to image quality, speed of result, seed placement, and memory consumption.

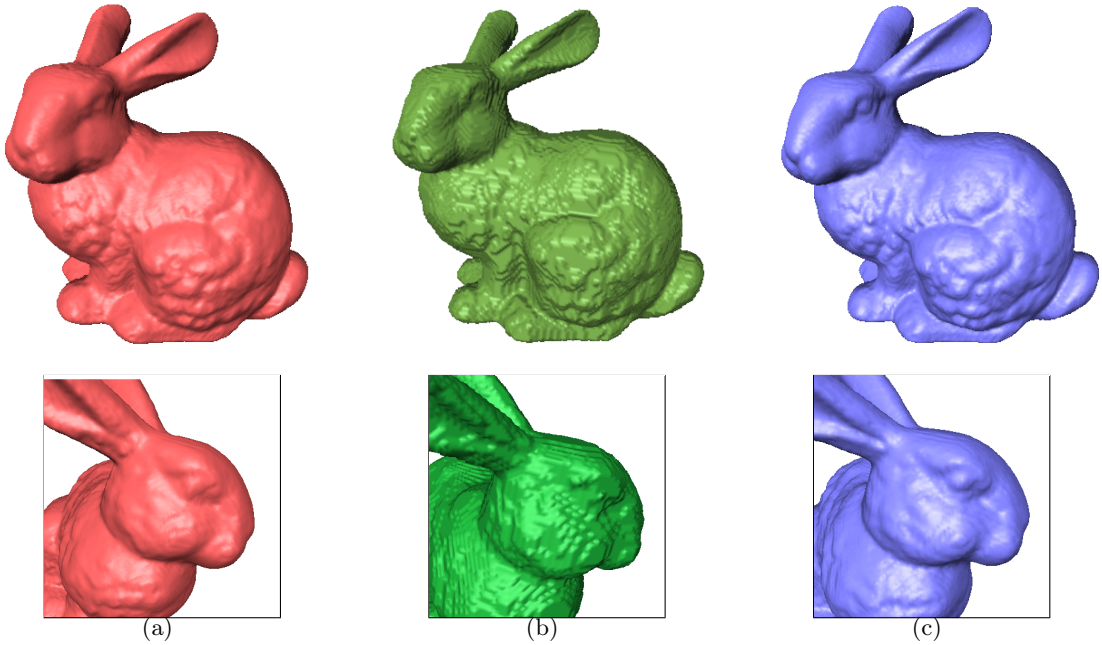


Figure 4.14: Grid size : $234 \times 297 \times 301$*, complete and zoomed results. (a) Total variation minimization result, (b) Graph cuts result, (c) power watershed result. Isosurfaces at 0.5 have been extracted on all results, and were downsampled by 2 to render the surfaces.*

Since then, we have also used power watershed for stereo-vision (Couprie, 2011) and sub-pixel accurate tangent surface estimation between touching grains (Jaquet et al., 2013). A reference implementation by Camille Couprie is freely available on the web at `http://www.esiee.fr/~coupriec/code.html`.

4.11 Conclusion and future work

We have described in this chapter the basis of many recent works in discrete optimization. We have started from basic network graphs and the maximum flow principles, and we have proceeded through the various algorithm and to the applications. We have shown how network graphs and maxflow-mincut algorithms can be used to respectively represent and optimize energies that are relevant for image analysis and computer vision. As examples, we have mostly shown image segmentation and image restoration, but the basic principles can be used in other applications such as stereo and tracking. We have shown that though the principles are fairly simple, the art of graph construction can be complex, due to its versatility. We have surveyed recent work on discrete flow-based image optimization methods.

Graph-cuts methods are interesting in particular because they allow practitioners to efficiently solve long-standing MRF formulations in computer vision, which even in the relatively recent past required much slower (e.g. simulated annealing) or much less effective (e.g. ICM) methods. They are also very interesting because formulations exist to solve widely useful energies, most notably some for which the data fidelity term has very few requirements. Many of these have strong guarantees on the solution, e.g. the final result is typically a local minimum,

within a known factor of the global solution. For convex priors, globally optimal solutions can be found. However, graph cuts are not able to solve all problems: sometimes a graph solution is not easy to propose, and many interesting energies are not submodular. Included in these are formulations involving so-called higher order cliques, i.e. triplets or more, which are sometimes difficult to optimize (Kolmogorov and Rother, 2007).

We also have presented some of our own contribution in discrete optimization. In particular, we presented quantized-convex moves for optimizing $\ell_2 - \ell_0$ sparsity measures, and a new, more general framework for discrete optimization, that includes, among others, watershed, graph cuts and random walkers. In this framework, a discrete-optimization version of the watershed was presented, termed the Power watershed. We showed applications to non-segmentation tasks, like image denoising and surface reconstruction.

In the next chapter, we will study some continuous formulations of flow methods and how they relate to optimization tasks.

Chapter 5

Continuous Flow-based methods for image segmentation and filtering

Based on the previous chapter, we have seen that discrete maximum flow/minimum cut methods are useful in optimization: as partitioning tools, image segmentation, but also in some specialized versions of linear and integer programming problem. These methods are classical although much research is still devoted to solving maxflow/mincut, in particular regarding still more efficient (e.g. parallel) algorithms. However, their efficient application to vision-related problems is quite recent.

Continuous versions of the maxflow/mincut problem have been studied as well, particularly by Iri (1979) and Strang (1983). More recently, Appleton (2004); Appleton and Talbot (2006) have proposed a continuous versions of flow-based methods. These have strong links to continuous Total Variation minimization considerations (Chambolle, 2004; Nikolova et al., 2006). We have also proposed a discrete calculus formulation of the problem (Couprie et al., 2011c), in order to reap benefits from both approaches.

Given the versatility and usefulness of graph cuts in the discrete domain, the question remains whether such techniques can be applied to the continuous domain as well. In particular, is there an equivalent to a flow in the continuous domain, and is there the equivalent of a maxflow-mincut theorem? We will see in this section that these questions can be answered in the affirmative.

5.1 Motivation

If graph cuts are so useful and effective in the discrete domain, why worry about the continuous domain at all? after all, the continuous domain is a model, where mathematical objects are not necessarily simple to deal with, and for which we will need to consider discretization again, should we wish to perform calculations on computers.

One motivation is of course the quest for understanding and generalization: our intuition, tools and techniques will be different in a different domain. Another is to realize that graph-cut models do exhibit some limitations. One is the grid bias, or lack of isotropy, which only gets worse as the dimensionality increases, and is difficult to deal with using graph cuts (Boykov and Kolmogorov, 2003). Applying graph cuts on segmentation and filtering problems leads to so-called *metrication* artifacts due to this bias. There is also always the hope that constraints are algorithmically different in a different domain. In particular, discrete maxflow/mincut

algorithms are difficult to parallelize effectively, but perhaps an equivalent algorithm in the continuous domain will behave differently.

Importantly, there are some deep links between discrete total variation minimization and maxflow-mincut optimization techniques. Continuous-domain as well as discrete-domain TV optimization has been an active area of research for several year for image restoration and other applications (Rudin et al., 1992; Vogel and Oman, 1996; Chambolle and Lions, 1997; Chan et al., 1999; Malgouyres, 2002b). It could be very useful to see if flow techniques in the continuum can be applied to solve similar problems.

5.2 Previous work

Hu and Gomory (see Hu (1969, Chapter 12)) provided some initial motivation for looking at the maxflow/mincut problem in the continuum, but sought to solve this problem using discrete tools. Initial theoretical investigations into maximum flow considerations in the continuous domain were performed by Iri (1979) and Strang (1983). Thorough mathematical analysis of both works have been proposed by Nozawa (1990, 1994) more recently. We now revisit these works, and provide links with Total Variation minimization.

5.2.1 Functions with bounded variation

In the remainder of this section, we consider functions with bounded variation on a domain Ω, which is an open subset of $\mathbb{R}^n$. We denote by $\overline{\Omega}$ its closure. Given a function f belonging to $L^1(\Omega)$, i.e. the measurable functions f of $\mathbb{R}^n$ that have a finite absolute value integral over Ω, its total variation TV is defined by

$$\mathrm{TV}(f,\Omega) = \sup \left\{ \int_\Omega f(x) \operatorname{div} \xi(x) \, \mathrm{d}\Omega : \xi \in C_c^1(\Omega, \mathbb{R}^n), |\xi(x)| \leq 1 \right\}. \tag{5.1}$$

Here, $C_c^1(\Omega, \mathbb{R}^n)$ is the set of continuously differentiable function of $\mathbb{R}^n$ with compact support included in Ω. Note that if $\overline{\Omega}$ is itself compact, this implies that f is zero almost everywhere on $\partial(\overline{\Omega})$. If not, f vanishes at infinity.

If $f \in C^1(\Omega)$, then the Green-Gauss formula yields:

$$\forall \xi \in C_c^1(\Omega, \mathbb{R}^n), \int_\Omega f \operatorname{div} \xi \, \mathrm{d}\Omega = - \int_\Omega \xi \nabla f \, \mathrm{d}\Omega. \tag{5.2}$$

The maximum is reached if $\xi = -\frac{\nabla f}{|\nabla f|}$ almost everywhere. In this case, we have

$$\mathrm{TV}(f,\Omega) = \int_\Omega |\nabla f| \, \mathrm{d}\Omega, \tag{5.3}$$

where the ∇ operator and the derivatives are understood in a weak sense. We have the following definition

Definition 5.1 (Space of bounded variations) *A function f of $L^1(\Omega)$ is said to have* bounded variation *on Ω if $\mathrm{TV}(f,\Omega) < \infty$. We define $\mathrm{BV}(\Omega)$ the space of all functions in $L^1(\Omega)$ with bounded variation.*

Note that the Sobolev space $W^{1,1}(\Omega)$ is a proper subspace of $\mathrm{BV}(\Omega)$. The converse is not true, for instance in 1-D in $\mathbb{R}$, a square function function (i.e. $f(x) = 1$ if $0 \leq x \leq 1, 0$ otherwise) is in $L^1(\mathbb{R})$, has bounded variation but is not in $W^{1,1}(\mathbb{R})$.

Many other details on functions of bounded variation and associated concepts can be found in Giusti (1984).

5.2.2 Iri's work

M. Iri (1979), and with applications in Taguchi and Iri (1982) proposed a purely continuous version of the maxflow-mincut theorem. Iri's presentation is entirely tensorial. In particular it allows capacities that varies with the direction of flow. His motivation is the approximation of dense and large discrete networks by continuous ones. Iri's mathematical description is formal but not very rigorous and was revisited by Nozawa (1990) in the Euclidean and scalar setting.

In Iri's framework, we considers a compact, n-dimensional Euclidean domain $\overline{\Omega}$, as well as two disjoint sets S and T belonging to the boundary $\partial\Omega$ of $\overline{\Omega}$. We denote the interior of $\overline{\Omega}$ by Ω, and we consider a flow $\mathbf{F}$ from S to T satisfying the following conditions:

$$\operatorname{div} \mathbf{F} = 0 \tag{5.4}$$

$$\mathbf{F}.\overrightarrow{n} = 0 \text{ on } \partial\Omega \backslash (S \cup T) \tag{5.5}$$

Here $\overrightarrow{n}$ is a unit outer normal vector to $\partial\Omega$. We assume $\partial\Omega$, S and T sufficiently smooth.

We further assume a non-negative function g on Ω and we formulate a maximum flow problem (MFPI) as follows:

$$\text{(MFPI) Maximize } \int_S \mathbf{F}.\overrightarrow{n}\, \mathrm{d}s \tag{5.6}$$

$$\text{subject to : } |\mathbf{F}| \leq g \text{ on } \Omega. \tag{5.7}$$

Here $\mathbf{F}$ is a vector flow from S to T and $\mathrm{d}s$ is a surface element.

Conversely, a cut C separating S and T is a subset of $\overline{\Omega}$ such that $S \subset \partial C$ and $T \cap \partial C = \emptyset$. We assume ∂C to be sufficiently smooth to be able do define the cost $K(C)$ and the dual minimum cost problem (MCPI) as follows:

$$\text{(MCPI) Minimize } K(C) = \int_{\partial C \cap \overline{\overline{\Omega}}} g\, \mathrm{d}s \tag{5.8}$$

$$\text{subject to : } C \text{ separates } S \text{ and } T. \tag{5.9}$$

Iri asserts that with sufficient smoothness, (MFPI) and (MCPI) are dual to each other, in the sense that the maximum flow of (MFPI) will be reached on the minimal cut of (MCPI). This is a very similar result to that of Ford and Fulkerson in the discrete case, however

this is somewhat restrictive because of the null divergence hypothesis in the whole of Ω. Sometime later, G. Strang independently proposed a more rigorous, more flexible framework for continuous maximum flows that we present in some details in the next section.

5.2.3 Strang's work on maximum flow through a continuous domain

In the discrete case, we considered in chapter 4 a flow network, i.e. a graph with edge capacities c_{ij}, endowed with a source s and a sink t. We considered a partition of this network into two regions S containing s and T containing t. The capacity of the cut is the sum of the capacity of the edges crossing the boundary between S and T:

$$\mathcal{K}(S,T) = \sum_{i \in S, j \in T} c_{ij} \tag{5.10}$$

Here (5.10) is the same as (4.6). Any flow F from s to t is constrained by this capacity as it must cross its associated boundary. We have seen that the maximum flow in the network is equal to the minimum cut among all possible S and T.

G. Strang (1983) considers a similar problem, in the 2-D continuous domain. Here the flow is a vector field defined in a compact domain $\overline{\Omega}$ of interior Ω, denoted $\mathbf{F} = (F_1(x,y), F_2(x,y))$ in a planar domain. Given a differentiable curve Γ, the flow across the curve is given by

$$|F|_\Gamma = \int_\Gamma \mathbf{F}.\overrightarrow{n}\mathrm{d}s, \tag{5.11}$$

where $\overrightarrow{n}$ is the unit normal to the curve. The capacity constraint is considered at each point to be defined by a scalar metric g. We have:

$$\|\mathbf{F}(x,y)\|_2 \leq g(x,y). \tag{5.12}$$

A different norm than the ℓ_2 norm can be considered, as well as a tensorial metric $\mathbf{g}$, however this is not crucial here. We considered a simply connected compact domain $\overline{\Omega}$ bounded by a single closed curve Γ. Here Γ is considered parametrized by a Lipschitz-continuous function, making the curve rectifiable and the parametrization differentiable. We assume the flow is constrained along Γ to be proportional to a given function φ such that:

$$\mathbf{F}.\overrightarrow{n} = \lambda\varphi \text{ on } \Gamma. \tag{5.13}$$

Here λ is an expression of the total scalar flow crossing Γ. Strang also specifies sources and sinks within $\overline{\Omega}$ by a function Φ, such that:

$$\operatorname{div}\mathbf{F} = \nabla.\mathbf{F} = -\lambda\Phi. \tag{5.14}$$

Here Φ and φ are positive for a sink representing respectively flow disappearing within Ω or leaving through Γ, and negative for a source, representing flow appearing within Ω or coming in through Γ. Using the divergence theorem, we have

$$\int_{\Omega} \nabla.\mathbf{F}\,\mathrm{d}\Omega = \oint_{\Gamma} \mathbf{F}.\overrightarrow{n}\,\mathrm{d}s \tag{5.15}$$

$$\int_{\Omega} -\lambda\Phi\,\mathrm{d}\Omega = \oint_{\Gamma} \lambda\varphi\,\mathrm{d}s, \tag{5.16}$$

and thus the conservation equation:

$$\int_{\Omega} \Phi\,\mathrm{d}\Omega + \oint_{\Gamma} \varphi\,\mathrm{d}s = 0 \tag{5.17}$$

5.2.3.1 Maximum flow through a domain

Given the above, Strang states the continuous maximum flow though a domain (CMFD) problem as follows:

(CMFD)

$$\text{Maximize } \lambda, \tag{5.18}$$

$$\text{Subject to: } \|\mathbf{F}\| \leq g, \tag{5.19}$$

$$\mathbf{F}.\overrightarrow{n} = \lambda\varphi, \tag{5.20}$$

$$\operatorname{div}\mathbf{F} = -\lambda\Phi. \tag{5.21}$$

We note that (5.20) and (5.21) are compatible due to the conservation laws. The system of equation and constraints has solutions, because any field $\mathbf{F}$ of the form $\mathbf{F} = (\psi_y, -\psi_x)$ for any C^2 function ψ that vanishes on Γ is a solution to both $\mathbf{F}.\overrightarrow{n} = 0$ and $\operatorname{div}\mathbf{F} = 0$, and so can be added to any solution to (5.20) and (5.21). We expect that for small λ, (5.19) can be satisfied for some $\mathbf{F}$, but for some larger λ it may not be. The problem is then to find the largest λ satisfying (5.19) to (5.21).

5.2.3.2 Minimum cut in a continuous domain

We are now interested to find an equivalent to the Ford-Fulkerson theorem in the continuous case. To define a cut we specify a new subset S of $\overline{\Omega}$ with finite boundary ∂S and the associated subset $T = \overline{\Omega}\backslash S$. As S may have some common boundary with $\overline{\Omega}$, we defined $\gamma = \partial S\backslash\Gamma$ to be the cut between S and T, and Γ_S the common part of ∂S with Γ. We have then $\partial S = \Gamma_S + \gamma$. We define the capacity of γ to be

$$C(S) = \int_{\gamma} g\,\mathrm{d}s \tag{5.22}$$

This capacity bounds the flow from S to T:

$$\int_S \operatorname{div} \mathbf{F} \, \mathrm{d}S = \int_{\partial S} \mathbf{F}.\overrightarrow{n} \, \mathrm{d}s \text{ (divergence theorem)}$$

$$\int_S -\lambda\Phi \, \mathrm{d}S = \int_{\Gamma_S} \mathbf{F}.\overrightarrow{n} \, \mathrm{d}s + \int_\gamma \mathbf{F}.\overrightarrow{n} \, \mathrm{d}s$$

$$\lambda \left| \int_S \Phi \, \mathrm{d}S + \int_{\Gamma_S} \varphi \, \mathrm{d}s \right| = \left| \int_\gamma \mathbf{F}.\overrightarrow{n} \, \mathrm{d}s \right| \leq C(S) \tag{5.23}$$

Relation (5.23) is one of weak duality and provides a bound on λ that depends only on S. Strang states the following theorem:

Theorem 5.2 (Continuous maxflow-mincut) *There exists some S for which the bound of* (5.23) *is tight, i.e. some maximum flow $\mathbf{F}$ and some set S for which*

$$\left| \int_\gamma \mathbf{F}.\overrightarrow{n} \, \mathrm{d}s \right| = \int_\gamma g \, \mathrm{d}s \tag{5.24}$$

In this case, the maximum flow is given by:

$$\max \lambda = \inf \frac{C(S)}{\left| \int_S \Phi \, \mathrm{d}S + \int_{\Gamma_S} \varphi \, \mathrm{d}s \right|} \tag{5.25}$$

The proof of this property in the general case where Φ is non-null is very interesting, as it involves duality with the concept of Total Variation.

5.2.3.3 Duality and Total Variation

To formulate the dual, we introduce the Lagrange multiplier $u \in \mathrm{BV}(\Omega)$, where $\mathrm{BV}(\Omega)$ is the space of functions with bounded variation as defined in section 5.2.1. The BV norm is called the Total Variation:

$$\|u\|_{\mathrm{BV}} = \int_\Omega |u| \, \mathrm{d}\Omega \tag{5.26}$$

We express the dual of the (CMFD) problem by transposing the divergence of $\mathbf{F}$ to the gradient of u and by transposing the norm. We formulate this as the Continuous Total Variation over a Domain minimization problem (CTVD), as follows:

$$(\mathrm{CMFD}^\star = \mathrm{CTVD}) \qquad \text{Minimize } \int_\Omega |\nabla u| g \, \mathrm{d}\Omega \tag{5.27}$$

$$\text{Subject to: } \int_\Omega u\Phi \, \mathrm{d}\Omega + \int_\Gamma u\varphi \, \mathrm{d}s = 1 \tag{5.28}$$

To see the link between (CMFD) and (CTVD), we consider the Total Variation of characteristic functions. Functions within BV do not have to be continuous, in fact the functions in BV are those whose first derivatives (in the sense of distributions) are finite measures. In particular, the characteristics function $u = \chi_S$ of a set S with finite perimeter has a BV-norm, equal to the perimeter of S. The characteristic of set S is a function which is uniformly 1 inside S and 0 outside. For such function, its weighted Total Variation is:

$$\int_\Omega |\nabla \chi_s| g \, \mathrm{d}\Omega = \int_\Gamma g \, \mathrm{d}s = C(S). \tag{5.29}$$

We now show that if we minimize TV over multiples of characteristic functions, the result will be a minimum cut. Multiples of characteristic functions are $u = \alpha\chi_S$. We have $\int_\Omega |\nabla u| g \, \mathrm{d}\Omega = \alpha C(S)$, as well as $\int_\Omega u\Phi \, \mathrm{d}\Omega = \alpha \int_S \Phi \, \mathrm{d}\Omega$ and $\int_\Gamma u\varphi \, \mathrm{d}s = \alpha \int_{\Gamma_S} \varphi \, \mathrm{d}s$. Overall, the α cancel each other out and this leads us to

$$\frac{\int_\Omega |\nabla u| g \, \mathrm{d}\Omega}{|\int_\Omega u\Phi \, \mathrm{d}\Omega + \int_\Gamma u\varphi \, \mathrm{d}s|} = \frac{C(s)}{|\int_S \Phi \, \mathrm{d}S + \int_{\Gamma_S} \varphi \, \mathrm{d}s|} \tag{5.30}$$

We can choose α such that $\int u\Phi \mathrm{d}\Omega + \int u\varphi \mathrm{d}s = 1$ to fulfill (5.28). Finally the left-hand side is precisely TV and the right-hand side is the minimum cut of (5.25).

5.2.3.4 Condition for optimality

Assuming $\mathbf{F}$ and u are feasible for both (CMFD) and (CTVD), we have simultaneously:

$$\begin{aligned} \nabla.\mathbf{F} &= -\lambda\Phi \\ \mathbf{F}.\overrightarrow{n} &= \lambda\varphi \\ |\mathbf{F}| &\leq g \\ \int_\Omega u\Phi \, \mathrm{d}\Omega + \int_\Gamma u\varphi \, \mathrm{d}s &= 1 \end{aligned}$$

Using the Gauss-Green formula:

$$\int_\Omega \mathbf{F}.\nabla u \, \mathrm{d}\Omega = -\int_\Omega u\nabla.\mathbf{F} \, \mathrm{d}\Omega + \oint_\Gamma u.\mathbf{F}\overrightarrow{n} \, \mathrm{d}s \tag{5.31}$$

By simple substitution we have on the one hand:

$$\int_\Omega \mathbf{F}.\nabla u \, \mathrm{d}\Omega = +\int_\Omega \lambda\Phi \, \mathrm{d}\Omega + \oint_\Gamma \lambda u.\varphi \, \mathrm{d}s = \lambda \tag{5.32}$$

On the other hand:

$$\mathbf{F}.\nabla u \leq |\mathbf{F}.\nabla u| \leq g|\nabla u| \tag{5.33}$$

Altogether, we have the weak duality:

$$\lambda = \int_{\Omega} \mathbf{F}.\nabla u \, \mathrm{d}\Omega \leq \int_{\Omega} g|\nabla u| \, \mathrm{d}\Omega \tag{5.34}$$

Optimality holds if the maximum of (CMFD) equals the minimum of (CTVD) and, whenever $|\nabla u|$ is non-zero:

$$\mathbf{F} = g\frac{\nabla u}{|\nabla u|} \tag{5.35}$$

5.2.3.5 Existence of a minimizing series of characteristic functions

There remain to justify why restricting u to the multiples of characteristics functions is sufficient. Strang does this using the coarea formula:

$$\int_{\Omega} |\nabla u| g \, \mathrm{d}\Omega = \int_{\mathbb{R}} \oint_{\gamma_t} g \, \mathrm{d}s \, \mathrm{d}t. \tag{5.36}$$

Here γ_t is the boundary of the set U_t, which is the binary set for which $u \geq t$. This formula expresses the total variation of any function it terms of the perimeters of all its level sets. For this formula to make sense, almost all are assumed to have finite length. Strang also uses and extends a result of Fleming (1954), which states that the extreme points of the unit ball in BV are reached by multiples of characteristics functions. Strang shows the existence (but not the general construction) of a minimizing, non-degenerate series u_n of characteristic functions converging to the optimum of both the (CMFD) and (CTVD) problems simultaneously.

We note that since the optimum is reached at a multiple of characteristic function, $|\nabla u|$ is zero almost everywhere, and that only on the boundary of this function is the optimality condition (5.35) reached.

Also, even though the presentation is done in 2D, Strang's results extend readily to a domain $\overline{\Omega}$ of arbitrary finite dimension, with Γ being of codimension 1.

More details about this proof, illustration, applications, etc can be found in Strang (1983). It is worth noting that Strang's work on continuous maxflow/mincut and the TV duality did not come with an algorithm for computing them practically.

5.2.4 The Nozawa conditions

S. Nozawa spent considerable efforts on Strang's continuous maflow/mincut proof to render it more formal. In the process he discovered some examples of duality gap between the (CMFD) and the (CTVD) formulations.

For an n-dimension problem, Nozawa found that Strang's theorem holds if the set of all feasible flows is a weak*-compact set in $L^\infty(\Omega, \mathbb{R}^n)$; the capacity must also be a bounded Borel

measurable function. The weak* topology, also called ultraweak topology, is related to the weak-topology operating on the set of bounded operators in $\mathbb{R}^n$, which is the coarsest topology that makes these operators continuous.

The most critical aspect of these condition is the one on the capacity g. Essentially, the capacity must be both bounded and continuous on the entire domain Ω. Nozawa goes on to provide counter-examples of duality gaps between the minimum cut and maximum flow solutions when g is not continuous or not bounded, both for the Iri and the Strang problems.

Note that since, following (5.30), the (CTVD) problem in Strang's 1983 paper is defined in terms of the perimeter of the minimum cut solution, the Nozawa results also provides examples of (CMFD) and (CDTV) duality gaps when the capacity g is non-continuous or non-bounded.

5.2.5 Total variation minimization algorithms

Since (CMFD) and (CTVD) are dual to one another, Strang's problem can be solved using either approach. Historically, the (CTVD) approaches seem more interesting to computer vision because Total Variation (TV) terms appear in many fundamental computer vision models, such as the Mumford-Shah functional (1989) of Equation (3.10), or the simpler Rudin-Osher-Fatemi (ROF) model (1992). A convex formulation is provided by Chan and Bresson (2010). Early ROF solutions based on level-sets can be seen as attempts to solve the (CTVD) problem.

More recently, A. Chambolle (2004) proposed a general-purpose TV minimization algorithm based on convex analysis. His algorithm is remarkable because it optimizes the ROF model exactly, and considers both the continuous and discrete points of view.

5.2.5.1 Chambolle's algorithm

To simplify notations, Chambolle considers an $N \times N$ 2D image. He denotes X the $\mathbb{R}^{N\times N}$ Euclidean space, and by Y the $X \times X$ vector space. He considers possibly the simplest discretization of the gradient $\nabla u = ((\nabla u)^1, (\nabla u)^2)$ of a function $u \in X$

$$(\nabla u)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \tag{5.37}$$

$$(\nabla u)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \tag{5.38}$$

Then the TV of u is given by

$$\mathrm{TV}(u) = \sum_{\substack{0\leq i\leq N \\ 0\leq j\leq N}} |\nabla u| \tag{5.39}$$

with $|.|$ the Euclidean norm in Y, i.e: $|y = (y_1, y_2)| = \sqrt{y_1^2 + y_2^2}$. With the scalar products in X and Y defined as the usual Euclidean ones, i.e:

$$\langle u, v\rangle_X = \sum_{i,j} u_{i,j} v_{i,j} \tag{5.40}$$

$$\langle p, q\rangle_Y = \sum_{i,j} p^1_{i,j} q^1_{i,j} + p^2_{i,j} q^2_{i,j} \tag{5.41}$$

We must have that the gradient and divergence are adjoint operators, meaning

$$\forall p \in Y \text{ and } \forall u \in X, \langle p, \nabla u\rangle_Y = \langle -\operatorname{div} p, u\rangle_X, \tag{5.42}$$

which implies the following discretization for the divergence operator:

$$(\operatorname{div} p)_{i,j} = \begin{cases} p^1_{i,j} - p^1_{i-1,j} & \text{if } 1 < i < N \\ p^1_{i,j} & \text{if } i = 1 \\ -p^1_{i-1,j} & \text{if } i = N \end{cases} + \begin{cases} p^2_{i,j} - p^2_{i,j-1} & \text{if } 1 < j < N \\ p^2_{i,j} & \text{if } j = 1 \\ -p^2_{i,j-1} & \text{if } j = N \end{cases} \tag{5.43}$$

In his paper, Chambolle starts from the definition of TV from (5.1) and defines as set K the closure of the set

$$\{\operatorname{div} \xi : \xi \in C^1_c(\Omega, \mathbb{R}^2), \forall x \in \Omega, |\xi(x)| \leq 1\} \tag{5.44}$$

Then, given a function $v \in X$ and $\lambda > 0$, he proposes an algorithm for solving

$$\underset{u \in X}{\text{minimize}} \frac{\|u - v\|^2}{2\lambda} + \text{TV}(u) \tag{5.45}$$

with $\|.\|$ the usual Euclidean norm in X, i.e. $\|u\| = \sqrt{\langle u, u\rangle}$. This is the TV-L2 problem, which can be interpreted as the Rudin-Osher-Fatemi (ROF) formulation for image denoising (Rudin et al., 1992).

Derivation of a fixed point algorithm If we consider a continuous version of the TV-L2 formulation (5.45):

$$\text{TVL2}(u) = \int_\Omega |\nabla u| \mathrm{d}\Omega + \lambda \int_\Omega (u - v)^2 \mathrm{d}\Omega \tag{5.46}$$

We are seeking to minimize this convex functional. Its Euler-Lagrange condition is,

$$\nabla . \left(\frac{\nabla u}{|\nabla u|} \right) + \frac{(u - v)}{\lambda} = 0 \tag{5.47}$$

This is classical from (Rudin et al., 1992). However, this equation is not valid in 0, so we replace $\frac{\nabla u}{|\nabla u|}$ by the vector $\mathbf{p}$ defined as

$$\mathbf{p}(x) = \begin{cases} \frac{\nabla u}{|\nabla u|} & \text{if } \nabla u(x) \neq 0 \\ \text{undefined} & \text{if } \nabla u(x) = 0 \end{cases} \tag{5.48}$$

This leads us to the following system:

$$-\nabla.\mathbf{p} + \frac{1}{\lambda}(u - v) = 0 \tag{5.49}$$

$$p|\nabla u| - \nabla u = 0 \tag{5.50}$$

A fixed point iteration for (5.50) is

$$\mathbf{p}^{n+1} = \mathbf{p}^n - \mathrm{d}t(\mathbf{p}^n|\nabla u| - \nabla u) \tag{5.51}$$

Solving for p^{n+1} yields:

$$\mathbf{p}^{n+1} = \frac{\mathbf{p}^n + (\tau/\lambda)\nabla u}{1 + (\tau/\lambda)|\nabla u|}, \tag{5.52}$$

by setting $\mathrm{d}t = \tau/\lambda$ with $\tau > 0$. To eliminate the dependence on u, we use (5.49) rewritten as:

$$u = \lambda\nabla.\mathbf{p} + v, \tag{5.53}$$

yielding:

$$\mathbf{p}^{n+1} = \frac{\mathbf{p}^n + \tau\nabla(\nabla.\mathbf{p}^n + v/\lambda)}{1 + \tau|\nabla(\nabla.\mathbf{p}^n + v/\lambda)|}, \tag{5.54}$$

This derivation is due to Unger et al. (2008). Chambolle shows more rigorously that a solution to (5.45) is given by

$$u = v - \pi_{\lambda K}(v), \tag{5.55}$$

where $\pi_{\lambda K}$ is the nonlinear projection onto the set λK. Chambolle then proposes the same fixed point algorithm as (5.54), with $0 < \tau < 1/8$ and $p^0 = 0$,

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau(\nabla(\operatorname{div} p_{i,j}^n - v/\lambda))_{i,j}}{1 + \tau|(\nabla(\operatorname{div} p^n - v/\lambda))_{i,j}|} \tag{5.56}$$

He then proves that $\lambda \operatorname{div} p^n$ converges towards $\pi_{\lambda K}(v)$ as $n \to +\infty$, and hence solves (5.45). Chambolle uses this algorithm to solve denoising problems in the presence of Gaussian noise, as well as an optimal zoom problem.

With this approach, we see that using a suitable function v and renormalization, we may be able to formulate and solve (5.30) in the case where $g \equiv 1$.

TV is a well-studied topic in image processing and computer vision. Weighted TV was studied by Vogel and Oman (1996). Generalized total variation was proposed by Bredies et al. (2010). Bresson and Chan (2008) proposed an efficient non-local version among many others.

5.2.6 Proximal methods

Interestingly, Chambolle's algorithm from the previous subsection can be interpreted as a method for providing a numerical solution for a general operator called the *proximity* operator, in the special case where the function of interest it the total variation.

Let $\Gamma^0(\mathbb{R}^n)$ be class of lower semicontinuous convex functions from $\mathbb{R}^n$ to $]-\infty,+\infty]$ such that $\operatorname{dom} f \neq \emptyset$. The following definition is due to Moreau (1965):

Definition 5.3 (Proximity operator) *Let $f \in \Gamma^0(\mathbb{R}^n)$. For every $x \in \mathbb{R}^n$, the minimization problem*

$$\underset{y\in\mathbb{R}^n}{\text{minimize}} \quad f(y) + \frac{1}{2}\|x-y\|^2 \tag{5.57}$$

has a unique solution, denoted prox_f*. The corresponding operator $\mathbb{R}^n \longrightarrow \mathbb{R}^n$ is the* proximity *operator of f.*

The proximity operator has many desirable properties for optimization algorithms. For instance, let ∂f be the set-valued subdifferential of f:

$$\partial f : \mathbb{R}^n \longrightarrow 2^{\mathbb{R}^n} : x \longmapsto \{u \in \mathbb{R}^n / (\forall y \in \mathbb{R}^n, (y-x)^\intercal u + f(x) \leq f(y)\}, \tag{5.58}$$

then

$$\forall (x,p) \in \mathbb{R}^n \times \mathbb{R}^n, p = \operatorname*{prox}_f x \Leftrightarrow x - p \in \partial f(p). \tag{5.59}$$

Other properties are also very useful. For instance the prox is firmly non-expansive, and its fixed point set is precisely the set of minimizers of f. These, and other useful properties, as well as many classical algorithms such as the Forward-Backward (Levitin and Polyak, 1966), Douglas-Rachford (1956) and Alternating Direction Method of Multipliers (ADMM, also known as Split-Bregman), initiated by Gabay (1983) are detailed in the tutorial of Combettes and Pesquet (2010).

We will be revisiting proximal methods at the end of the next section, when we introduce a general primal-dual method after presenting a method for computing maximum flows in the continuous domain.

5.3 Continuous maximum flows

We have seen that TV is a useful concept in computer vision for image restoration. It can be used as a prior in many optimization frameworks. However, its dual, the maximum flow, has not been studied as much.

In Appleton and Talbot (2003), we proposed a method for computing continuous maximum flows (CMF), which we detail briefly here.

5.3.1 A continuous flow simulation

The continuous maximum flow algorithm consists of simulating the following constrained flow:

$$\frac{\partial P}{\partial t} = -\nabla \cdot \mathbf{F} \tag{5.60}$$

$$\frac{\partial \mathbf{F}}{\partial t} = -\nabla P \tag{5.61}$$

$$\text{s.t.} \quad \|\mathbf{F}\| \leq g, , \tag{5.62}$$

where $\mathbf{F}$ is a vector field simulating velocity, P is a scalar field simulating pressure, and g is a scalar constraint field, interpreted in the remainder as a metric. Moreover, we define a source s and a sink t, both separate arbitrary regions of the space. We impose the boundary conditions $P_s = 1$ and $P_t = 0$ without loss of generality.

We can interpret this system of equations as a continuous analogue of the preflow-push algorithm of Goldberg and Tarjan (1988) for the discrete maxflow-mincut problem, described in chapter 4.5. We showed in Appleton and Talbot (2006) that this flow simulation, at convergence, resolved to:

$$\begin{array}{lll} \nabla \cdot \mathbf{F} & = 0 & \\ \nabla P & = 0 & \text{if } |\mathbf{F}| < g \\ \nabla P & = -\lambda \mathbf{F} \quad \text{where} \quad \lambda \geq 0 & \text{if } |\mathbf{F}| = g. \end{array} \tag{5.63}$$

From this, we showed that P is monotonically decreasing from s to t, and so all thresholds of P define a binary partition separating s from t. In addition, all the isosurfaces of P are minimal surfaces for the metric g. Our article (Appleton and Talbot, 2006) is reprinted as Appendix G in this manuscript.

So this system of constrained PDEs solves the *continuous maximum flow* through a domain (CMFD) problem defined by Strang, as reprinted in section 5.2.3. In the following, we termed it the AT-CMF formulation.

A finite difference implementation is relatively straightforward and can be used to solve the same problem as graph cuts, with some key differences:

- Relatively easy parallelisation
- Absence of grid bias.

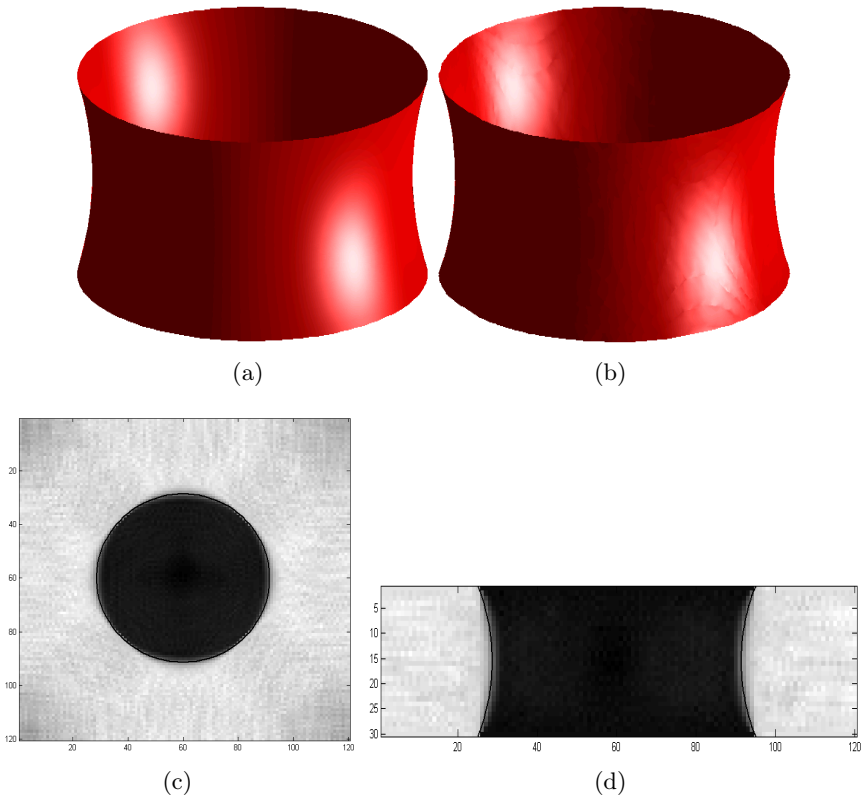
An example of system solution is given in Fig. 5.1. In this, the source consists of two disks of equal radius situated on the top and bottom horizontal surface of a parallelipedic box, while the sink consists of the four vertical sides of the same box. The metric g is set to uniformly 1. Since the metric is uniform, we expect a geometric minimal surface as a result, and this is indeed what we achieve. The measured average error between the expected analytical and the actual result was less than 0.11 pixel.

5.3.2 Relation to TV minimization

From section 5.2.3, we know that TV and maximum flows are related. It is interesting to draw the relation at the algorithmic level.

We can rewrite (5.48) involving the variable $\mathbf{p}$ as follows:

$$|\nabla u| = \max_{\|\mathbf{p}\| \leq 1} (\mathbf{p}.\nabla u) \tag{5.64}$$



(a) (b)

(c) (d)

Figure 5.1: The catenoid test problem. (a) The expected minimal surface, constructed analytically. (b) The iso-surface at value 0.5 of P at convergence. (c) A horizontal slice through P. The expected cross-section is overlayed in black. (d) A vertical slice through P. The expected cross-section is overlayed in black.

Substituting into (5.46), we have:

$$\mathrm{TVL2}(u) = \max_{\|\mathbf{p}\|\leq 1} \left\{ \int_{\Omega} \mathbf{p}.\nabla u \, \mathrm{d}\Omega + \frac{1}{2\lambda} \int_{\Omega} (u-v)^2 \mathrm{d}\Omega \right\}. \tag{5.65}$$

The divergence theorem states that

$$\int_{\Omega} \mathbf{p}.\nabla u = -\int_{\Omega} u \nabla.\mathbf{p}; \tag{5.66}$$

The Euler-Lagrange condition for minimizing (5.65) then becomes

$$-\nabla.\mathbf{p} + \frac{1}{\lambda}(u-v) = 0, \text{, and thus:} \tag{5.67}$$

$$u = \lambda \nabla.\mathbf{p} + v \tag{5.68}$$

which is the same as (5.49). With it, we can rewrite (5.65) as

$$\min(\text{TVL2}(u)) = \max_{\|\mathbf{p}\|\leq 1} \left\{ -\int_\Omega \lambda(\nabla.\mathbf{p})^2 \mathrm{d}\Omega - \int_\Omega v\nabla.\mathbf{p}\,\mathrm{d}\Omega + \frac{1}{2\lambda}\int_\Omega \lambda^2(\nabla.\mathbf{p})^2 \mathrm{d}\Omega \right\} \tag{5.69}$$

$$= \min_{\|\mathbf{p}\|\leq 1} \left\{ \int_\Omega v\nabla.\mathbf{p}\,\mathrm{d}\Omega + \frac{\lambda}{2}\int_\Omega (\nabla.\mathbf{p})^2 \mathrm{d}\Omega \right\}. \tag{5.70}$$

This latter formulation is free of u, and its Euler-Lagrange, valid everywhere is:

$$-\nabla(v + \lambda\nabla.\mathbf{p}) = 0, \|\mathbf{p}\| \leq 1. \tag{5.71}$$

A way to optimize (5.65) is therefore to propose a constrained fixed-point solution for (5.71):

$$\overline{\mathbf{p}}^{n+1} = \mathbf{p}^n + \frac{\tau}{\lambda}(\nabla(v + \lambda\nabla.\mathbf{p}^n)) \tag{5.72}$$

$$\mathbf{p}^{n+1} = \frac{\overline{\mathbf{p}}^{n+1}}{\max(1, |\overline{\mathbf{p}}^{\mathbf{n+1}}|)}. \tag{5.73}$$

This is equivalent to a classical projected gradient algorithm. Together with a fixed-point discretization of (5.68), we have the system:

$$\overline{\mathbf{p}}^{n+1} = \mathbf{p}^n + \frac{\tau}{\lambda}(\nabla u^n) \tag{5.74}$$

$$u^{n+1} = v^n + \lambda\nabla.\mathbf{p}^n \tag{5.75}$$

$$\mathbf{p}^{n+1} = \frac{\overline{\mathbf{p}}^{n+1}}{\max(1, |\overline{\mathbf{p}}^{\mathbf{n+1}}|)}. \tag{5.76}$$

We see that by substituting P for u and $\mathbf{F}$ for $\mathbf{p}$, and by taking $g \equiv 1$, then this system is exactly a discretization of equations (5.60)–(5.62).

In other words, the AT-CMF formulation is equivalent to a projected gradient scheme for optimizing TV-L2. A slight modification of the above projected gradient scheme, given by Unger et al. (2008), allows for arbitrary positive scalar field g. In the same work, it is shown that the AT-CMF/projected gradient scheme is significantly faster than the Chambolle fixed point algorithm, at least at segmentation tasks.

5.3.3 Generalization to primal-dual convex optimization

Chambolle and Pock (2011) proposed a first-order primal-dual algorithm for solving non-smooth convex problems. The formulation as a saddle-point minimization problem is as follows:

$$\min_{\alpha\in A}\max_{\beta\in B} \{\langle H\alpha, \beta\rangle + \Phi(\alpha) - \Psi^*(\beta)\}, \tag{5.77}$$

where A and B are finite-dimensional real vector spaces, H is a linear operator $H : A \longrightarrow B$, Φ and Ψ^* are two proper, convex, lower-semicontinuous functions: $\Phi : A \longrightarrow \mathbb{R} \cup \{+\infty\}$, $\Psi^* : B \longrightarrow \mathbb{R} \cup \{+\infty\}$. Ψ^* is the Legendre-Fenchel convex conjugate of a lower semicontinuous function Ψ.

Algorithm 4: A general primal-dual scheme for solving 5.77

Data: Scalar variables $\tau > 0$, $\sigma > 0$, $\theta \in [0, 1]$
Initial vector variable $\alpha^0 \in A$, $\beta^0 \in B$ and $\overline{\alpha}^0 = \alpha^0$
Result: Converged primal variable α and β
for $j = 1$ *to* J **do**
$\beta^{j+1} = (I + \sigma \partial \Psi^*)^{-1}(\beta^j + \sigma H \overline{\alpha}^j)$
$\alpha^{j+1} = (I + \tau \partial \Phi)^{-1}(\alpha^j - \tau H^* \beta^j)$
$\overline{\alpha}^{j+1} = \alpha^{j+1} + \theta(\alpha^{j+1} - \alpha^j)$

An algorithm for solving (5.77) consists of a gradient ascent step on the dual variable β together with a resolvent operator. For the primal variable α a gradient descent step is performed together with another resolvent operator. An extra gradient step is applied to the variable α. Timesteps τ and σ are chosen so that $\tau\sigma L^2 < 1$, with L the Lipschitz constant of H such that $L^2 = \|H\|^2$.

The resolvent operators can be written as proximity operators (5.57) (Bauschke and Combettes, 2011, Chap. 23):

$$\beta = (I + \sigma \partial \Psi^*)^{-1}(\overline{\beta}) = \underset{\beta}{\operatorname{argmin}} \left\{ \Psi^*(\beta) + \frac{1}{2\sigma} \|\beta - \overline{\beta}\|^2 \right\} \tag{5.78}$$

$$= \underset{\sigma\Psi^*}{\operatorname{prox}}(\overline{\beta}) \tag{5.79}$$

$$\alpha = (I + \tau \partial \Phi)^{-1}(\overline{\alpha}) = \underset{\alpha}{\operatorname{argmin}} \left\{ \Phi(\alpha) + \frac{1}{2\tau} \|\alpha - \overline{\alpha}\|^2 \right\} \tag{5.80}$$

$$= \underset{\tau\Phi}{\operatorname{prox}}(\overline{\alpha}) \tag{5.81}$$

This algorithm can be improved if either or both of Ψ^* and Φ are uniformly convex, i.e. with a Lipschitz continuous gradient. Much more details can be found in Chambolle and Pock (2011), further work in Pock and Chambolle (2011), as well as related work in Pock et al. (2010). The thesis of M. Unger (2012) is also of high didactic interest.

Our AT-CMF algorithm of section 5.3.1 can be seen as an early, more primitive instance of the Chambolle-Pock (CP) algorithm. Indeed applying the CP algorithm to the TVL2 problem or equivalently the ROF formulation, yields exactly the same projected gradient algorithm at that of section 5.3.2 above. This is demonstrated by Unger (2012, chap.2).

Other related, primal-dual, prox-based approaches have been proposed recently in the literature, for instance by Combettes and Pesquet (2012) or Condat (2012). In other words, this area of research is currently very active.

5.4 Convex optimization on graphs

In the previous sections, we have shown how flow methods in image processing relate to, and in some cases improve upon more established solutions for inverse problem solving in imaging.

However, in spite of significant progress, there are some areas where improvements are necessary. First is the notion that discrete and continuous approaches to optimization are different and incompatible. In fact, they have different strengths. As we have shown in Chaux et al. (2010), reprinted as Appendix F in this manuscript, using both continuous and discrete optimization together can be fruitful.

Here we would like to extend this notion by providing a formulation of arbitrary convex problems using continuous convex methods on graphs. We start with a formulation of CMFs on graphs, which we extend to a more general framework.

5.4.1 Continuous maximum flows on graphs

As before, we consider a graph $G = (V, E)$ with vertices $v \in V$ and edges $e \in E$. Specifically we consider a transport graph G with two additional nodes s and t, respectively a source and sink. The cardinalities are given by $n = |V|$ and $m = |E|$. An edge e spanning two vertices v_i and v_j is denoted e_{ij}. In this section, we will consider weights on both edges and nodes. The weight of a node v_i is denoted g_i, while the weight of an edge e_{ij} is denoted $\tilde{g}_{ij}$. We denote $\mathbf{g}$ the vector of $\mathbb{R}^n$ that contains the g_i for all the vertices, and $\tilde{\mathbf{g}}$ the vector of $\mathbb{R}^m$ that contains the $\tilde{g}_{ij}$ for all the edges. We define a flow through edge e_{ij} as F_{ij}, where $F_{ij} \in \mathbb{R}$. The vector $\mathbf{F} \in \mathbb{R}^m$ denotes the flow through all the edges in the graph. The flow is oriented, meaning that a positive flow on edge e_{ij} indicates that the direction of flow is from v_i to v_j, while a negative flow indicates a direction from v_j to v_i.

We use the incidence matrix A as a key operator for defining a graph-based formulation of continuous maximum flows. Specifically $A \in \mathbb{R}^{m \times n}$ defines the equivalent of the gradient operator, while $A^\intercal$ corresponds to the divergence operator. This the definition given in discrete calculus (Grady and Polimeni, 2010) is:

$$A_{e_{ij}v_k} = \begin{cases} +1 & \text{if } i = k \\ -1 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \tag{5.82}$$

An example of incidence matrix is given in Fig 5.2.

5.4.1.1 Graph-based formulation of the CMF problem

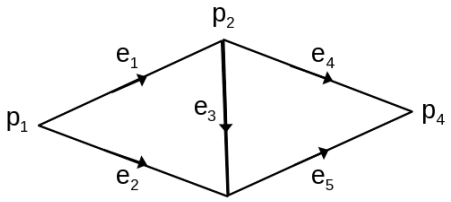
Strang's simplified formulation of the CMFD problem can be expressed as:

$$\max F_{st} \tag{5.83}$$

$$s.t. \quad \nabla . \vec{F} = 0 \tag{5.84}$$

$$\|\vec{F}\| \leq \mathbf{g}. \tag{5.85}$$

The maximisation must be understood in the sense more precisely defined in section 5.2.3.1. The conservation equation (5.84), where $\vec{F}$ is a vector field, is true everywhere in the domain



Incidence matrix $A \in \mathbb{R}^{M \times N}$

$$A = \begin{array}{c|cccc} & p_1 & p_2 & p_3 & p_4 \\ \hline e_1 & -1 & 1 & 0 & 0 \\ e_2 & -1 & 0 & 1 & 0 \\ e_3 & 0 & -1 & 1 & 0 \\ e_4 & 0 & -1 & 0 & 1 \\ e_4 & 0 & 0 & -1 & 1 \end{array}$$

Figure 5.2: The incidence matrix of a small graph

excepted source and sink. We have seen how a continuous formulation can solve this problem in section 5.3.1.

The purely discrete, classical maximal flow formulation on graph can be written in the following way:

$$\max F_{st} \tag{5.86}$$
$$s.t. \mathbf{A}^\intercal \mathbf{F} = 0 \tag{5.87}$$
$$|\mathbf{F}| \leq \tilde{\mathbf{g}}. \tag{5.88}$$

This is equivalent to the discrete formulations of chapter 4. Now to express the CMF problem on graph, we propose to find a graph-equivalent for these three equation. We will still be maximizing F_{st}, so this does not change. Following Grady and Polimeni (2010) and Elmoataz et al. (2008), we can express a continuous vector field on the vertices of a lattice, so the conservation equation (5.87) does not change, although its interpretation changes (from a discrete to a vector field). Finally, we can follow the same articles to define the ℓ_2 norm of the flow field as $\sqrt{|\mathbf{A}^\intercal|\mathbf{F}^2}$, where $|\mathbf{M}|$ denotes a matrix formed by taking the absolute value of each element individually, where $\mathbf{M}^2$ is a matrix formed by taking the square of each element individually, and so is the square root $\sqrt{\mathbf{M}}$. Given this, we formulate the problem as follows:

$$\max F_{st} \tag{5.89}$$
$$s.t. \quad \mathbf{A}^\intercal \mathbf{F} = 0 \tag{5.90}$$
$$|\mathbf{A}^\intercal|\mathbf{F}^2 \leq \mathbf{g}^2 \tag{5.91}$$

In our article Couprie et al. (2011c) we termed this formulation *Combinatorial Continuous Maximum Flows* (CCMF). The key differences with the purely discrete formulation are that we are dealing with a vector flow, and the capacity constraint. In the purely discrete case, the capacity is expressed on edges. In our case, it is expressed on vertices. This is illustrated on Fig. 5.3.

5.4.1.2 Dual formulation

It is possible to mechanically (but not trivially!) derive the Lagrangian dual formulation of CCMF, yielding:

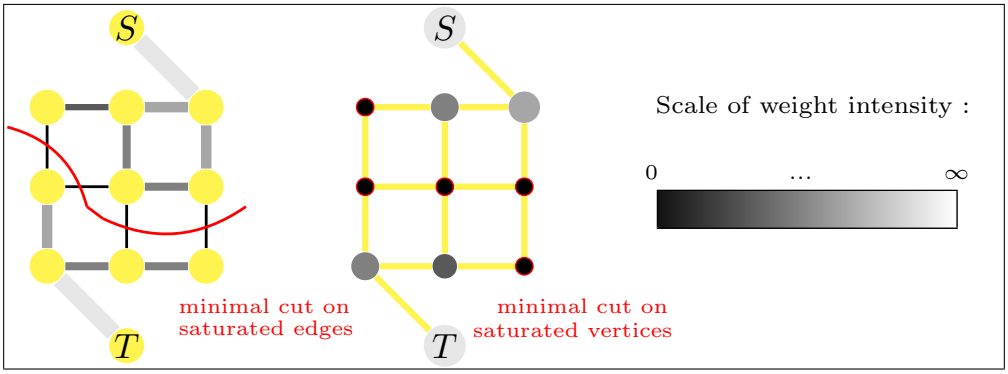


Figure 5.3: The difference between classical max-flow on a graph with the combinatorial continuous max-flow (CCMF) on a graph is that classical max-flow uses edge-weighted capacities while CCMF uses node-weighted capacities. This difference is manifest in the different solutions obtained for both algorithms and the algorithms required to find a solution. Specifically, the solution to the CCMF problem on a lattice does not exhibit metrication bias.

$$\begin{aligned} \min_{\lambda\in\mathbb{R}^n,\ \nu\in\mathbb{R}^n} \quad & \lambda^\intercal \mathbf{g}^2 + \tfrac{1}{4}\Big(\mathbf{1}^n \cdot /(|\mathbf{A}|\lambda)\Big)^\intercal \Big((c+\mathbf{A}\nu)^2\Big), \\ \text{s. t.} \quad & \lambda \geq 0, \end{aligned} \tag{5.92}$$

This is equivalently written in (5.95), and the optimal solution $(\mathbf{F}^*, \lambda^*, \nu^*)$ verifies

$$\max_{\mathbf{F}} \mathbf{c}^\intercal \mathbf{F} = \mathbf{c}^\intercal \mathbf{F}^* = 2\lambda^{*\intercal}\mathbf{g}^2, \tag{5.93}$$

and the n following equalities

$$\lambda^* \cdot |\mathbf{A}^\intercal|\Big((c+\mathbf{A}\nu^*)\cdot /(|\mathbf{A}|\lambda^*)\Big)^2 = 4\lambda^* \cdot \mathbf{g}^2. \tag{5.94}$$

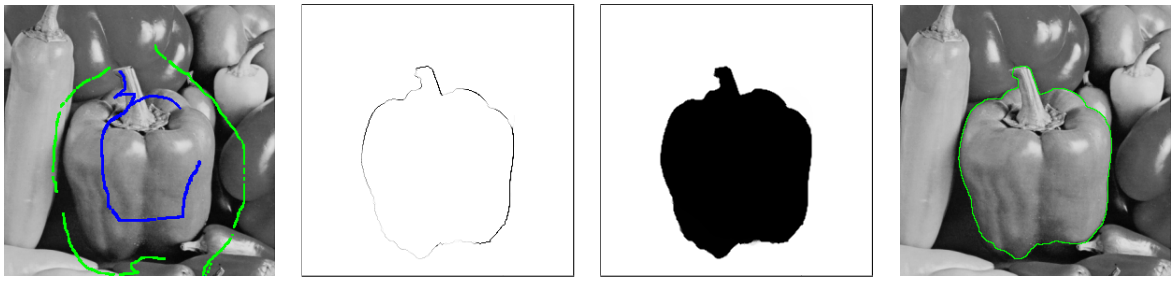
The expression of the CCMF dual may be written in summation form as

$$\min_{\lambda,\nu} \sum_{v_i\in V} \overbrace{\lambda_i g_i^2}^{\text{weighted cut}} + \overbrace{\frac{1}{4}\sum_{e_{ij}\in E\setminus\{s,t\}} \frac{(\nu_i-\nu_j)^2}{\lambda_i+\lambda_j}}^{\text{smoothness term}} + \overbrace{\frac{1}{4}\frac{(\nu_s-\nu_t-1)^2}{\lambda_s+\lambda_t}}^{\text{source/sink enforcement}} \tag{5.95}$$
$$\text{s. t. } \lambda_i \geq 0 \quad \forall i \in V.$$

Interpretation: The optimal value λ^* is a weighted indicator of the saturated vertices (a vertex v_i is saturated if $|\mathbf{A}^\intercal|_i \mathbf{F}^2 = g_i^2$ where $|\mathbf{A}^\intercal|_i$ indicates the ith row of $|\mathbf{A}^\intercal|$):

$$\lambda^*(v_i) \begin{cases} > 0 & \text{if } |\mathbf{A}^\intercal|_i \mathbf{F}^2 = \mathbf{g}(v_i)^2, \\ = 0 & \text{otherwise.} \end{cases} \tag{5.96}$$

The variables ν_s and ν_t are not constrained to be set to 0 and 1, only their difference is constrained to be equal to one, however without loss of generality we can impose this constraint af-



(a) (b) λ (c) ν (d) Threshold ν at .5

Figure 5.4: The dual problem to CCMF is a node-weighted minimum cut in which the variable λ is a weighted indicator vector labeling boundary nodes and the variable ν is a nearly binary vector indicating the source/sink regions. As a result, the contours of ν are slightly blurry. This is due to the equilibrium effect between the two dual variables. In practice, as λ is nonzero only in presence of a contour, ν is binary almost everywhere, except on a very thin line.

ter the fact. The term ν is at optimality a weighted indicator of the source/sink/saturated vertices partition:

$$\nu^*(v_i) = \begin{cases} 0 \quad \text{if } v_i \in S, \\ \text{a number between (0) and (1)} \quad \text{if } |\mathbf{A}^\intercal|_i \mathbf{F}^2 = \mathbf{g}(v_i)^2, \\ 1 \quad \text{if } v_i \in T. \end{cases}$$

The expression (5.93) of the CCMF dual shows that the problem is equivalent to finding a minimum weighted cut defined on the nodes.

Finally, the "weighted cut" is recovered in (5.95), and the "smoothness term" is compatible with large variations of ν at the boundary of objects because of a large denominator (λ) in the contour area. An illustration of optimal λ and ν on an image is shown on Fig. 5.4.

5.4.1.3 Solving the CCMF problem

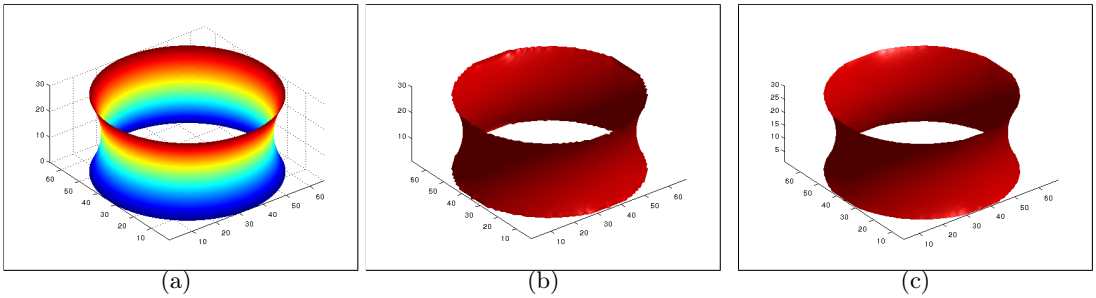
The formulation of the CCMF problem is convex, including the constraints set. We were able to formulate a primal and a dual, so a reasonable way to solve this problem is to use a primal-dual interior point method (PDIP). This is fairly unusual in imaging, since interior point methods have a reputation of becoming slow and memory hungry as the dimensionality of the problem increases.

However the systems to solve are large but very sparse, and our PDIP implementation in Matlab was able to perform adequately, including solving segmentation problems for large 3D images. For more details, please refer to Couprie et al. (2011c).

5.4.1.4 Results

We were able to show that CCMF perform well compared with various formulations of TV and graph cuts for segmentation problems. Most importantly, it is indeed free of metrication artifacts, and converges in very few iterations.

In Fig 5.5, we show that we obtained even better results than AT-CMF on the catenoid test.



(a) (b) (c)

Figure 5.5: The catenoid test problem: The source is constituted by two full circles and sink by the remaining boundary of the image. (a) Surface computed analytically, (b) isosurface of P obtained by AT-CMF, (c) isosurface of ν obtained by CCMF. The root mean square error (RSME) has been computed to evaluate the precision of the results to the surface computed analytically. The RSME for AT-CMF is 1.98 and for CCMF 0.75. The difference between those results is due to the fact that the AT-CMF algorithm enforces exactly the source and sink points, leading to discretization around the disks. In contrast, the boundary localized around the seeds of ν is smooth, composed of grey levels. Thus the resulting isosurface computed by CCMF is more precise.

One main advantage of the CCMF formulation is that it converges very quickly compared with iterative solvers for TV or CMF, in the case of segmentation. Either require relatively few iteration for denoising, but to achieve a strong binary partition in the presence of weak edges may take many iterations and a long time. In 5.6, we show an example of segmentation on an artificial problem. CCMF converges quickly and reliably, typically in a few tens of iterations, even for difficult problems.

More results, including CCMF computed on a graph that is not a lattice, standard speed tests and result comparisons can be found in Couprie et al. (2011c). Our algorithm is amenable to parallelisation and GPU implementations, which would be a necessity for large problems.

This success with convex formulation on graphs lead us to consider solving more general inverse problems in a similar fashion.

5.4.2 Dual-constrained regularization on graphs

So far we only have considered the segmentation problem. This is a specific inverse problem seeking to restore (or detect) the contour of objects present in the image. More generally, we can consider an image restoration problem, where the "true" data has been corrupted by noise, blur, motion, etc. Although the two may seem quite different, in reality they can be formulated in very similar ways. Specifically, a weighted TV model can be appropriate. Let us consider the following formulation:

$$\underset{u}{\text{minimize}} \int_{\Omega} \Big(\int_{\Omega} w_{x,y}(u_y - u_x)^2 dy \Big)^{1/2} dx + \frac{1}{2\lambda} \int_{\Omega} (u_x - v_x)^2 dx, \tag{5.97}$$

where v is a degraded image, u the restored image, x and y arbitrary points of the domain Ω, and $w_{x,y}$ an arbitrary non-negative weight defined on Ω^2. As shown in Chan et al. (1999),

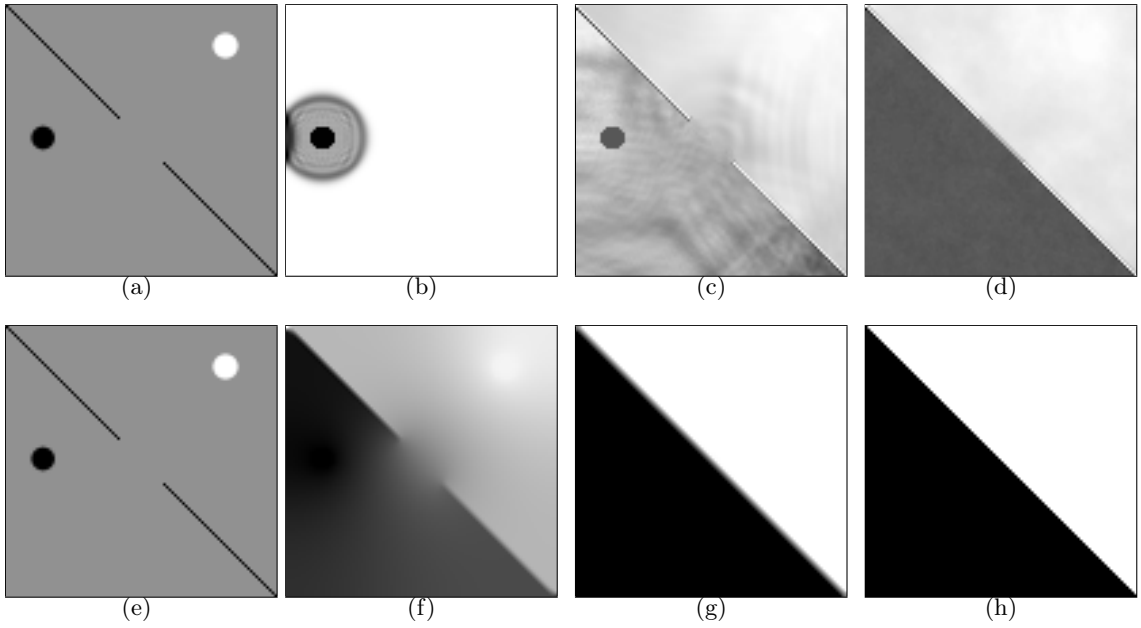


Figure 5.6: Segmentation of an artificial image with AT-CMF (top row) and CCMF (bottom row). Top row (AT-CMF): (a) Image where the black and white discs are seeds. AT-CMF result stopped in (b) after 100 iterations, (c) 1000 iterations, (d) 10000 iterations. Bottom row (CCMF): (e) Image where the black and white discs are seeds, CCMF result ν after (f) 1 iterations, (g) ν after 15 iterations iterations and (h) threshold of the final ν.

this problem is equivalent to the following min-max problem:

$$\underset{u}{\text{minimize}}\Big(\max_{||p||_\infty \leq 1} \int_{\Omega^2} w_{x,y}^{1/2}(u_y - u_x)p_{x,y}\mathrm{d}y\,\mathrm{d}x + \frac{1}{2\lambda}\int_{\Omega}(u_x - v_x)^2\mathrm{d}\Omega,\Big) \tag{5.98}$$

where p is a two-variable function and

$$\|p\|_\infty = \sup_{x\in\Omega}\Big(\int_{\Omega} p_{x,y}^2 dy\Big)^{1/2}.$$

In the discrete framework, this formulation becomes, following Bougleux et al. (2007) and Gilboa and Osher (2007)

$$\underset{u\in\mathbb{R}^n}{\text{minimize}} \sum_{i\in J}\Big(\sum_{j\in N_i} w_{i,j}(u_j - u_i)^2\Big)^{1/2} + \frac{1}{2\lambda}\sum_{i=1}^{n}(u_i - v_i)^2 \tag{5.99}$$

The dual formulation, which can be optimized by a projection algorithm (Gilboa and Osher, 2007), and expressing the gradient operator with the adjacency matrix $\mathbf{A}$ is given by

$$\underset{u\in\mathbb{R}^n}{\text{minimize}}\Big(\max_{||p||_\infty \leq 1} p^\intercal\big((\mathbf{A}u)\cdot\sqrt{w}\big) + \frac{1}{2\lambda}\|u - v\|^2\Big), \tag{5.100}$$

By introducing a vector $F \in \mathbb{R}^m$ with components $F_{i,j}$ such that, for every $i \in J$ and $j \in N_i$, $\mathbf{F}_{i,j} = p_{i,j}\sqrt{w_{i,j}}$, the problem can be reformulated as

$$\underset{u\in\mathbb{R}^n}{\text{minimize}}\Big(\max_{\mathbf{F}\in C} \mathbf{F}^\intercal \mathbf{A}u + \frac{1}{2\lambda}\|u-v\|^2\Big), \tag{5.101}$$

where

$$C = \{(\mathbf{F} \in \mathbb{R}^m \mid (\forall i \in J) \sum_{j\in N_i} \frac{\mathbf{F}_{i,j}^2}{w_{i,j}} \leq 1\}. \tag{5.102}$$

Finally, we extend the fidelity part by allowing (1) a linear degradation operator H, which can correspond to a blur, motion, a wavelet decomposition (Malgouyres, 2002a) or even a tomography reconstruction operator (Chouzenoux et al., 2013b); (2) a more complex correlation matrix Λ in the form of a positive definite matrix instead of a single Lagrangian scalar, and (3) we consider a more general convex constraint set C.

$$\underset{u\in\mathbb{R}^n}{\text{minimize}}\Big(\sup_{\mathbf{F}\in C} \mathbf{F}^\intercal \mathbf{A}u + \frac{1}{2}(\mathbf{H}u-v)^\intercal \Lambda^{-1}(\mathbf{H}u-v)\Big), \tag{5.103}$$

where $v \in \mathbb{R}^q$ is the observed vector of data, $\mathbf{H} \in \mathbb{R}^{q\times n}$ is the linear degradation operator and Λ is a weighting symmetric definite-positive matrix in $\mathbb{R}^{q\times q}$. We termed this the Dual-Constrained Total Variation (DCTV) restoration model.

5.4.2.1 Example of interesting class of constraint sets

The proposed optimization approach allow us to address nonempty convex sets C that can be decomposed as an intersection of closed convex subsets $(C_r)_{1\leq r\leq s}$ of $\mathbb{R}^m$, the projections onto which take closed forms. An example of a set C of interest is given by

$$\begin{aligned} &C = \bigcap_{r=1}^{s} C_r \\ &(\forall r \in \{1,\dots,s\}) \qquad C_r = \{F \in \mathbb{R}^m \mid (\forall i \in S_r)\ \|\theta^{(i)} \cdot F\|_\alpha \leq g_i\}. \end{aligned} \tag{5.104}$$

where $(S_r)_{1\leq r\leq s}$ is a partition of $\{1,\dots,n\}$,

$\|\cdot\|_\alpha$ is the ℓ_α norm of $\mathbb{R}^m$ with $\alpha \in [1,+\infty]$ and, for every $i \in \{1,\dots,n\}$, $\theta^{(i)} \in]0,+\infty[^m$ is a vector of multiplicative constants for every couple of node (i',j) with $i' \in J$ and $j \in N_{i'}$.

The form of C in (5.104) includes (5.102) as a particular case where $\alpha = 2$ and

$$\begin{aligned} &(\forall i \in \{1,\dots,n\}) \quad (\forall i' \in J)(\forall j \in N_{i'})\ \ \theta^{(i)}_{i',j} = \begin{cases} \frac{1}{\sqrt{w_{i',j}}} & \text{if } i' = i \text{ and } j \in N_{i'} \\ 0 & \text{otherwise} \end{cases} \\ &\qquad\qquad g_i = 1. \end{aligned} \tag{5.105}$$

However, the proposed approach offers significantly more flexibility.

In this section, we are mainly interested in the case when for every $i \in \{1,\dots,n\}$, $\theta^{(i)}$ is the i-th line vector of $|\mathbf{A}^\intercal|$. Specifically, C may be defined as

$$C = \{\mathbf{F} \in \mathbb{R}^m \quad | \quad g^{\cdot 2} - |\mathbf{A}^\intercal|\mathbf{F}^{\cdot 2} \in [0,+\infty[^n\}, \tag{5.106}$$

where $g = (g_i)_{1 \leq i \leq n}$ and, for every vector a, $a^{\cdot 2} = a \cdot a$.

The constraint given in the convex set of (5.106) also appears in the Combinatorial Continuous Maximum Flow (CCMF) problem of section 5.4.1.1 (Couprie et al., 2011c). The problem studied here may be seen as an extension of the CCMF problem (applied to clustering problems in graphs, such as image segmentation) to multi-label problems.

Concerning the choice of the node weight g_i at the i-th node, a simple strategy consists of considering a monotonically decreasing function of the data gradient. More specifically, given positive reals ϵ and χ, we suggest using g defined as

$$(\forall i \in \{1, \ldots, n\}) \qquad g_i = \exp(-\chi \|\nabla \overline{x}_i\|_2) + \epsilon, \tag{5.107}$$

where $\overline{x}$ is some reference data defined on a graph, for instance image data, which corresponds to some rough estimate of x, and

$$\|\nabla \overline{x}_i\|_2 = \begin{cases} (\sum_{j \in N_i} (\overline{x}_i - \overline{x}_j)^2)^{\frac{1}{2}} & \text{if } i \in J \\ 0 & \text{otherwise} \end{cases} \tag{5.108}$$

corresponds to the Euclidean norm of its discrete gradient $\nabla \overline{x}_i$ at node i. In the absence of a contour, g_i takes large values, so are the components of F corresponding to nonzero values of $\theta^{(i)}$, preventing large local variations of x in the minimization (5.103). Conversely, in the presence of a contour, $g_i \simeq \epsilon$, and the components of F corresponding to nonzero values of $\theta^{(i)}$ are small, thus allowing large local variations of x. In image filtering applications, in addition to intensities information, g may be used to penalize changes in other relevant image quantities such as color or texture.

5.4.2.2 Proposed algorithms for DCTV

It is possible to solve Problem (5.103) efficiently by proximal methods (Combettes and Pesquet, 2010).

To do so, we define the support function σ_C of the closed convex constraint set C as

$$\sigma_C \colon \mathbb{R}^m \to]-\infty, +\infty] \colon a \mapsto \sup_{\mathbf{F} \in C} \mathbf{F}^\intercal a. \tag{5.109}$$

This is a proper lower-semicontinuous convex function, the conjuguate of which is the indicator function of C,

$$\imath_C \colon F \mapsto \begin{cases} 0 & \text{if } F \in C, \\ +\infty & \text{otherwise.} \end{cases} \tag{5.110}$$

This leads us to consider the following optimization problem:

$$\min_u \sigma_C(\mathbf{A}u) + \frac{1}{2}(\mathbf{H}u - v)^\intercal \Lambda^{-1} (\mathbf{H}u - v) + \frac{\eta}{2} \|\mathbf{K}u\|^2, \tag{5.111}$$

where $\eta \in]0, +\infty[$ and $\mathbf{K} \in \mathbb{R}^{n \times n}$ is the projection matrix onto the nullspace of $\mathbf{H}$, specifically, $\mathbf{K} = I - \mathbf{H}^\intercal (\mathbf{H}\mathbf{H}^\intercal)^{-1} \mathbf{H}$. When $\mathbf{H}$ is injective ($\operatorname{rank} \mathbf{H} = n$), the last term vanishes and (5.111) is strictly equivalent to (5.103). The term $x \mapsto \eta \|\mathbf{K}u\|^2/2$ thus aims at introducing an additional regularization when $\mathbf{H}$ is not injective, so that the objective function remains strictly convex.

The following holds:

Property 1 *Problem* (5.111) *admits a unique solution* $\widehat{u}$*. The dual Fenchel-Rockafellar form of the problem is*

$$\min_{F} \varphi(\mathbf{F}) + \imath_C(\mathbf{F}), \tag{5.112}$$

where $\varphi\colon \mathbf{F} \mapsto \frac{1}{2}\mathbf{F}^\intercal \mathbf{A}\Gamma\mathbf{A}^\intercal\mathbf{F} - \mathbf{F}^\intercal\mathbf{A}\Gamma\mathbf{H}^\intercal\Lambda^{-1}f$ *and* $\Gamma = (\mathbf{H}^\intercal\Lambda^{-1}\mathbf{H} + \eta\mathbf{K})^{-1}$*. The optimal solution to the primal problem* (5.111) *is deduced from any optimal solution* $\widehat{\mathbf{F}}$ *of the dual problem by the relation*

$$\widehat{u} = \Gamma\left(\mathbf{H}^\intercal\Lambda^{-1}v - \mathbf{A}^\intercal\widehat{\mathbf{F}}\right). \tag{5.113}$$

Not every projection algorithm will be suitable to solve this problem, since the projection onto C is not explicit in general. Also, the operator $\mathbf{A}\Lambda\mathbf{A}^\intercal$ is singular in general. In order to numerically solve (5.112), we note that $C = \bigcap_{r=1}^{s} C_r$, so that $\imath_C$ can be decomposed into the sum of the indicator functions of the convex subsets $(C_r)_{1\leq r\leq s}$. Hence, the problem is equivalent to solving

$$\underset{\mathbf{F}\in\mathbb{R}^m}{\text{minimize}} \sum_{r=1}^{s} \imath_{C_r}(\mathbf{F}) + \varphi(\mathbf{F}). \tag{5.114}$$

Parallel Proximal Algorithm (PPXA) The above sum of $(s+1)$ convex functions can be efficiently optimized by resorting to the Parallel ProXimal Algorithm (PPXA) proposed by Combettes and Pesquet (2008).

As shown in Algorithm 5, this requires computing in parallel projections onto each set C_r with $r \in \{1, \ldots, s\}$, which are defined $\forall F \in \mathbb{R}^m$ as $\mathsf{P}_{C_r}(\mathbf{F}) = \underset{\Phi\in C_r}{\operatorname{argmin}} \|\Phi - \mathbf{F}\|$. Note that the convergence of the sequence $(F_k)_k$ generated by this algorithm to a solution $\widehat{F}$ of (5.112) is guaranteed, which allows us to deduce a solution to (5.111) by using Relation (5.113).

Algorithm 5: Parallel proximal algorithm solving (5.114)

Fix $\gamma > 0$ and $\nu \in]0, 2[$. Set $k = 0$.
Choose $y_{1,0} = y_{2,0} = \ldots = y_{s+1,0} \in \mathbb{R}^m$ and F_0.
repeat
  for $r = 1, \ldots, s+1$ **do in parallel**
    $\pi_{r,k} = \begin{cases} \mathsf{P}_{C_r}(y_{r,k}) & \text{if } r \leq s \\ (\gamma A\Gamma A^\intercal + I)^{-1}(\gamma A\Gamma H^\intercal\Lambda^{-1}v + y_{s+1,k}) & \text{otherwise} \end{cases}$
  $z_k = \frac{2}{s+1}(\pi_{1,k} + \cdots + \pi_{s+1,k}) - F_k$
  for $r = 1, \ldots, s+1$ **do in parallel**
    $y_{r,k+1} = y_{r,k} + \nu(z_k - p_{r,k})$
  $F_{k+1} = F_k + \frac{\nu}{2}(z_k - F_k)$
  $k = k + 1$
until *convergence*

Note that even for simple problems, the projection onto C may not be explicit. For instance in the case of a 4-connected lattice, this projection must be decomposed. Note also that the computation of $\pi_{s+1,k}$ requires a matrix inversion. In the case where $\mathbf{H}$ and $\mathbf{\Lambda}$ are (or can

be well approximated by) circulant-block circulant matrices, and the graph is regular, this step can be performed efficiently, for example by following De Mazancourt and Gerlic (1983); Fischer and Modersitzki (1999).

An alternative formulation is desirable in cases we do not want to solve any linear system at all. For this we may propose using a primal-dual scheme such as the one proposed by Chambolle and Pock (see Algorithm 4). This is essential when the graph is not regular, for instance for non-local regularization. Note however that *if* the graph is regular, then our PPXA formulation is usually much faster in spite of the matrix inversion. For more details, please refer to our much more detailed article (Couprie et al., 2011d).

5.4.2.3 Results

Here we only show a subset of the more interesting results achievable with our proposed approach

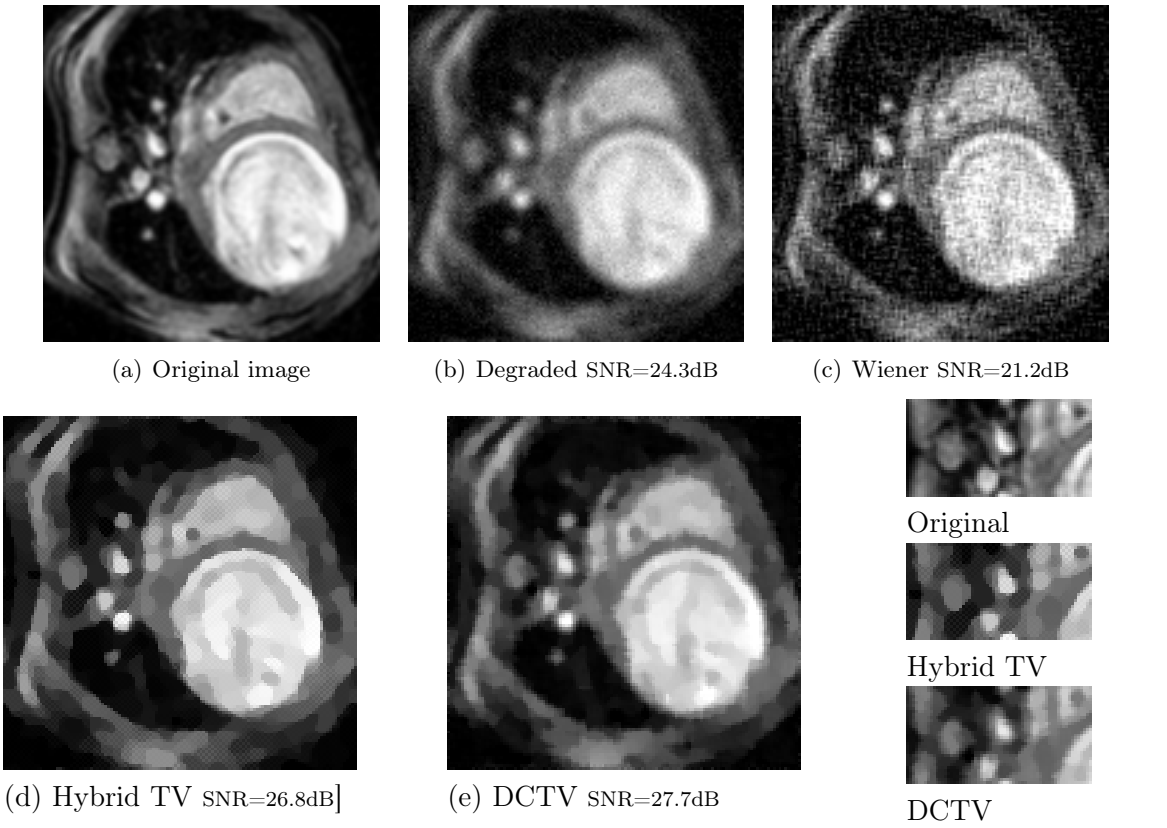
Non-local denoising On Fig. 5.7, we show some pure denoising results. Rather than using locally connected graphs, non-local strategies (Buades et al., 2005; Bresson, 2009) have been shown to achieve denoising improvements. A non-local strategy may naturally be employed in the DCTV framework. The weights between non-neighbor nodes are computed following the main idea of Buades et al. (2005): for each pixel p of the image, the squared sum of differences (SSD) between the intensities in a block around p and all other blocks in a large neighborhood around that block is computed. Then, edges are added between p and the nodes producing the best matchs, with weights corresponding to the normalized SSD scores. Since the Laplacian of this non-local graph is no longer circulant block circulant, we propose to employ the a primal-dual scheme in this case. Figure 5.7 presents a non local DCTV result obtained using this strategy compared with a non local TV result. About 15 iterations are necessary for primal-dual scheme to produce this result, taking about two seconds using a C parallel implementation on a 2.5GHz 8-core Xeon system used previously. The projections on the different (9 in the example of Figure 5.7) convex sets are performed in parallel. By comparison, the specific non local TV implementation by Bresson (2009) takes four iterations to converge in about one second. Although more flexible, our implementation is therefore competitive in this case.

Deblurring We now give some quantitative comparison results for joint denoising and deblurring tasks.

We compare the DCTV results to Wiener based deconvolution using the Matlab function ‘‘`deconvwrn`’’. The comparison also includes the hybrid TV/wavelet regularization method of Combettes and Pesquet (2008). We report in Fig. 5.8 the result of the restoration of images corrupted with additive white zero-mean Gaussian noise with variance $\sigma^2 = 5$, and $\sigma^2 = 10$, and convolved with uniform blur kernels of size 5×5 and 7×7. We observe that DCTV unsurprisingly outperforms the standard Wiener filter. More importantly, DCTV is competitive with a state-of-the-art method proposed by Combettes and Pesquet (2008), both quantitatively in term of SNR, and qualitatively, without presence of checkerboard artifacts observed for the Hybrid TV method in Figure 5.8. Furthermore, the results are obtained twice as fast, in the same number of iterations, using a Matlab implementations for both methods.

Figure 5.7: Local and nonlocal denoising (Gaussian noise of variance $\sigma^2 = 20$). The nonlocal weights are computed using Bresson (2009). Regularization parameters used to obtain the best results: (e) $\lambda = 0.048$ (f) $\lambda = 0.093$.

Mesh denoising Since the dual-constrained TV-based formulation is defined on arbitrary graphs, u is not limited to represent only image pixel values. In Fig. 5.9, we present an example of mesh denoising, where $\bar{u}$ is a vector composed of the spatial coordinates $\bar{u}_X$, $\bar{u}_Y$ and $\bar{u}_Z$ of the mesh nodes. In this experiment, we added a randomly oriented noise vector with zero-mean white Gaussian magnitudes to the original node coordinates of a mesh. This results



(a) Original image (b) Degraded SNR=24.3dB (c) Wiener SNR=21.2dB

(d) Hybrid TV SNR=26.8dB] (e) DCTV SNR=27.7dB

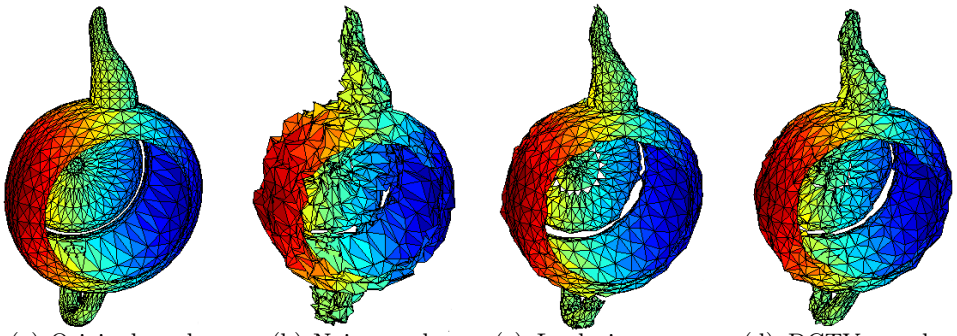
Figure 5.8: Denoising and deblurring an MRI image corrupted with synthetic Gaussian 5×5 *blur and Gaussian noise (*$\sigma^2 = 10$*).*
We observe in the TV hybrid result the presence of checkerboard artifacts due to the use of discrete filters for approximating the gradient. DCTV reduces the staircase effect of TV while preserving more details. Parameter used: for the Hybrid TV with regularization parameters (Combettes and Pesquet, 2008) $\alpha = 0$ and $\beta = 0.025$, and for DCTV with $\lambda = 0.005$ and $\eta = 0.04$.

in noisy mesh nodes of coordinates v_X, v_Y and v_Z. The degradation model is the following:

$$v = \begin{bmatrix} v_X \\ v_Y \\ v_Z \end{bmatrix} = \begin{bmatrix} \bar{u}_X \\ \bar{u}_Y \\ \bar{u}_Z \end{bmatrix} + \sigma^2 \begin{bmatrix} b_X \\ b_Y \\ b_Z \end{bmatrix}, \tag{5.115}$$

where b_X, b_Y, b_Z represent uncorrelated vectors of additive noises with unit magnitude variance.

This application shows that the DCTV framework is also well suited for regularizing this type of data. We compare our denoising result to a typical mesh smoothing technique, and show that DCTV outperforms Laplacian smoothing.



(a) Original mesh (b) Noisy mesh (c) Laplacian regularization (d) DCTV regularization

RMSE = 0.161 RMSE = 0.142 RMSE = 0.139

Figure 5.9: Example of mesh denoising using DCTV on the spatial coordinates of the nodes. We use the Root Square Mean Error to measure an average distance to the original mesh. (c) Laplacian smoothing (d) The M+SFBF algorithm was used to optimize DCTV with $\lambda = 0.25$. The Root Mean Square Error measures a distance between the result and the original mesh.

5.5 Extensions

We have extended existing variational restoration models by generalizing the constraint on the projection variable of the dual TV formulation. This new approach shows improved results compared with the weighted TV approach in image restoration applications. The advantage of our DCTV formulation over unconstrained formulations is that it allows us to take into account both edges and node weights, as well as more general constraints on the flow variable. As shown in our experiments, this behaviour results in better contrast preservation in addition to SNR improvements. Secondly, our dual formulation allows us to employ fast optimization strategies without requiring approximations. Also, the versatility of DCTV makes it appropriate in a large variety of contexts.

More generally, the proposed proximal algorithms make it possible to efficiently solve convex minimization problems involving the support function of an intersection of convex sets as a penalty term. It is also worth emphasizing that this approach can be applied to any graph data structures, in particular those frequently employed in 3D modeling.

In this manuscript, we have not presented a series of extensions that we have worked on:

- More advanced noise models. Implicitly, our data fidelity terms assume white additive Gaussian noise. This is appropriate for a variety of contexts, but it is possible to achieve better results in cases where the noise is more difficult to handle. Recently, we have presented some interesting results in the presence of the much more flexible Poisson-Gauss noise model. This was the result of the PhD work of Anna Jezierska (Jezierska et al., 2012; Jezierska, 2013).

- GPU implementations. Because these restoration tasks can be time consuming, and they are frequently highly parallelizable, taking the time to accelerate them can be fruitful. This was the topic of the PhD work of László Marak (Marak, 2012).

- Because the operator $\mathbf{H}$ can be any linear operator, it has immediate applications to

image reconstruction, such as MRI or X-Ray tomography. We have begun work with tomography operators using these and similar techniques (Chouzenoux et al., 2013b).

- Non-convex regularization: some of these operators extend to the non-convex case (Jezierska et al., 2011; Chouzenoux et al., 2011, 2013a). So far we have mostly worked with smooth regularization terms, but non-smooth ones will be investigated in the near future.

5.6 Conclusion and future work

We have surveyed recent work on flow-based image optimization methods in the last two chapters. We have described the purely discrete case, a solution to continuous case involving flow simulation and finite differences, and finally the discrete calculus case. All three cases have interesting properties:

- The discrete case (graph cut) is the simplest problem, equivalent to a linear program, that efficient specific algorithms can solve. The continuous case (Continuous Max Flows) and discrete calculus cases (Combinatorial Continuous Maximum Flows) are more complex. Their solution is isotropic not biased by the grid. Their corresponding problem is still convex, but solving them requires significantly more sophisticated tools.
- The AT-CMF formulation provides efficient algorithm, which is parallel and particularly efficient in 3D. However, its properties are only beginning to be understood. Convergence can be slow and under its PDE formulation, is not mathematically guaranteed, in spite of good practical results. The fact that it can be interpreted as a projected gradient algorithm does provide some convergence properties.
- In contrast, CCMF and by extension DCTV are more tightly defined mathematically. Convergence is guaranteed. The best algorithm so far for CCMF is based on a primal-dual interior-point method. It exhibit quite reasonable results in 2D, but in 3D the problem quickly becomes too dense and solving it with a direct solver requires too much memory in practice.
- DCTV extends the formulation of flow-based constraints to image and more generally graph-based restoration tasks. With it comes the flexibility of graph formulations, as well as the effectiveness of projection-based algorithms, including primal-dual approaches to avoid inverting linear systems when they become too dense and/or irregular.

At the end of this long chapter, we see that studying these problems requires considering a vast array of methods, all with their particularities, cases where they are efficient and cases where they are not. This is essentially the state of the art in optimization methods for imaging today. We have contributed to advancing the state of the art in terms of what is achievable today, but much remains to be done for these techniques to become easier to handle, and to apply them to a larger family of real problems.

One area we would like to extend these methods is the blind deconvolution case, where the PSF or blur operator is not known but must be discovered along with the restored image. This is a non-convex problem with many practical applications.

We would also like to study the case where the object to be restored or segmented is thin and has a particular connected structure, for instance blood vessel networks or aerial trees in

the lung. This is also a problem that is challenging to formulate and solve. There is a growing interest in this area.

Chapter 6

General conclusion and future work

It would seem natural to conclude on this body of work with some personal remarks, and also outline future projects. In this section I will switch away from the academic "we".

6.1 Past work

In this document I have tried to put into perspective a significant portion of what I have done in image analysis research over the last 20 years. I am not sure I have succeeded in producing a digestible, let alone interesting document, so I must thank you, the reader, for making it to that point[1].

In presenting work related to thin objects and more recent work in optimization, I have skipped over lots of work, perhaps with smaller impact than what I have presented. One of the joys of working in imaging is the possibility of seeing one's work applied to many different fields. It is also a great source of distraction, and so contrary to many of my colleagues, I unfortunately cannot claim to have a unified theme to my research. This has made writing this document more difficult and perhaps less interesting to read to any single person.

Science does not, perhaps similarly to a morphological path, proceed along a straightforward predictable way. I personally came to science via some detours, and even though I've immensely enjoyed making some contributions, it seem to have taken a long time for me to become reasonably proficient at it, at least in a way that I'm not completely ashamed of.

It has been interesting to me to look backward and realize that the very first problem I looked at in my scientific career, the problem of segmenting thin objects, had been so important. I do not think I have solved it. I am under the impression that a lot of work is still needed.

I continually feel the need to contribute to areas of image processing / image analysis that have some interesting applications, which means that most of my research has been project-driven. It would seem natural to look at the prospective projects to see what is to come. Of course I can only present what I have in mind at the present time, not what will actually happen.

[1] Unless you are reading this first of course...

6.2 Current planned work

Since this manuscript is supposed to present work worthy of an "habilitation", this means I have every intention to continue working in the area of image processing/image analysis and so on. This is a selection of topics I'm planning to work on in the near future:

6.2.1 Segmentation and medical imaging

A general purpose segmentation method still does not exist. With new student Eloïse Grossiort and with long-term collaborators Laurent Najman and Nicolas Passat, we will be investigating segmenting and measuring the volume of cancer lesions in Positron Emission Tomography. This is a collaboration with Michel Meignan and Emmanuel Itti at Hôpital Henri-Mondor, and will involve the company Keosys, a French SME developing medical imaging software.

6.2.2 Joint segmentation, reconstruction and restoration

Image restoration is a major application of inverse problems. Our approaches, both convex and non-convex, have a lot of potential in the field of tomography. With Saint-Gobain Recherche we are exploring the possibility of scaling current algorithms to the enormous sizes (typically $2048 \times 2048 \times 2048$ voxels) handled by practitioners working in synchrotron facilities such as the European Synchrotron Radiation Facility (ESRF). Our optimization approaches have a lot of potential for handling cases that are difficult for back-propagation or even so-called algebraic, iterative methods. Among those are complex noise statistics and continuous acquisition leading to wedge-type convolutions. Non-convex regularization terms and non-Gaussian fidelity terms are going to be a necessity to succeed.

6.2.3 Discrete transforms

Perhaps tangentially related to other works, I'm currently involved with Yukiko Kenmochi and Nicolas Passat in the supervision of Phuc Ngo (Ngo et al., 2012), We are studying the combinatorial aspects of discrete transforms, in particular the possibility of defining incremental transforms changing as few pixels at a time, in order to define a discrete, continuous path in transform space. The combinatorial aspect yields very large size problems, but the computation is relatively straightforward. We are working towards adding topological constraints and proposing conditions for transformed objects to remain topologically invariant under discrete rigid transforms.

6.2.4 Vivabrain

The VIVABRAIN project is an ANR project recently accepted, involving applied mathematicians, image analysts, neurologists and the company Kitware, who produces the open-source projects VTK and ITK, respectively a visualization toolkit and an imaging (sorry, "insight") toolkit, both widely used in medical imaging.

The objective is to propose a complete pipeline for the segmentation, delineation and topological correction of brain blood vessels, leading to the simulation of blood flows in anatomically correct blood networks. In turn these simulations will allow us to produce phantoms that will be used to assess the quality of 3d image filtering and segmentation methods. All the software

and models will be open-source. Validated portions will eventually become part of VTK and ITK, which should be very interesting.

This projects involves Université Paris Est, Université de Reims, Université de Strasbourg, the Institut National Polytechnique de Grenoble and Kitware. 5 PhD students including 3 funded by this ANR project will work on this project, which has started in January 2013.

6.3 Unplanned future work

Prediction is very difficult, especially about the future[2], however I hope to continue working in imaging-related projects. One topic is to approach thin object filtering and segmentation from an optimization point of view. Some proposals already exist, particularly involving non-local and wavelet operators, but I feel more work is needed, particularly in 3D. I will continue to be interested in efficient algorithms. One interesting way explored by Markus Unger in his thesis (Unger, 2012) is to incorporate some ideas from discrete optimization like swap moves in continuous optimization algorithms, to dramatically speed up convergence. However this is not applicable to all contexts. With Ania Jezierska we have been working on some of these aspects and we hope to develop them in the near future.

Another area of investigation is to pursue the idea of incorporating some curvature information in variational models.

Overall, it appears that only a minority of researchers seem to be convinced that both discrete and continuous approaches bring interesting aspects and solutions to a particular problem. The discrete calculus framework, recently popularized by Grady and Polimeni (2010) is one way both aspects can be combined, but it is surely not the only way. I hope to continue contributing both to continuous and discrete approaches in the future.

[2]Niels Bohr, 1885-1962.

Part III

Appendix

Content of the appendix

This appendix contains the following documents:

1. My CV and list of publications;
2. A short description of 2 industry-related projects; and
3. 5 significant publications

Appendix A

CV of Hugues Talbot

Details:

Date of Birth:	11 Octobre 1966
Location:	Parthenay, Deux-Sèvres, France
Nationalities:	French and Australian (acquired in 2001)
Family:	Maried, 2 children (09/2000 & 02/2006)

Education:

1993	PhD from the École des Mines de Paris. Dissertation on "Morphological analysis of man-made vitreous fibres". Co-supervised by professors Jean Serra, Dominique Jeulin (ENSMP) and Linn W. Hobbs (Massachusetts Institute of Technology), with highest honours.
1990	Master's degree, Université Paris-VI, Artificial intelligence and pattern recognition, with honours.
1989	Engineering degree, majoring in computer science, Ecole Centrale de Paris.

Prizes and scholarships:

2006	DuPont prize for the SolarScan project, with Polartechnics Ltd
2005	Australian society of engineers awards for the Solarscan project
2004	Research prize, CSIRO-MIS.
1990-1993	CIFRE PhD scholarship, with Isover Saint Gobain.
1984-1989	Higher education scholarship from the French ministry of education.

Professional experience:

2004-today	Associate professor (with tenure), ESIEE, Laboratoire d'Informatique Gaspard-Monge (LIGM), Algorithms, Architecture, Analysis and Image Synthesis(A3SI) team, France.
1994-2004	Principal Research Scientist, CSIRO Mathematical and Information Sciences Biotech Imaging Group (Sydney, Australia).
1990-1994	Research engineer, Isover Saint Gobain (Fontainebleau and Rantigny, France).
1991-1992	Visiting Scientist and Research Associate, Massachusetts Institute of Technology, (Cambridge, Mass., USA).
1989	Software Developer, Gattégno Conseil, (Sceau, France).
1988-1989	Consultant, Automatisme et Avenir Informatique, (Châtenay-Malabry, France).

Other activities:

2001-present	Associate editor of the journal *Image analysis and stereology*.
2004	Associate lecturer at Sydney University, ELEC-4302. Image processing and image analysis.
2002-2004	Secretary of the Australian Pattern Recognition Society (APRS).
2002-present	Member and secretary of the steering committee of the International Society for Mathematical Morphology.

Langages :

Anglais	bilingual
German	working knowledge

Skills

Image processing, image analysis

- Mathematical morphology; segmentation; partial differential equations; texture analysis; medical imaging; computer vision; motion analysis; stereovision, linear, convex and combinatorial optimization; linear and non-linear filtering.

Supervision

- Current PhD candidates:

  1. Ali Kanj, co-supervised with Jean-Christophe Pesquet, Université Paris-Est Marne-la-Vallée, on a CIFRE scholarship with Sublab Inc, – due end of 2016
  2. Odyssé Merveille, co-supervised with Nicolas Passat, Université de Reims, on an ANR VIVABRAIN scholarship, – due to graduate end of 2016;
  3. Éloise Grossiord, co-supervised with Laurent Najman, LIGM, ESIEE, on a CIFRE scholarship with Keosys Inc, – due end of 2016;
  4. Olivia Miraucourt, co-supervised with Stéphanie Salmon, laboratoire de mathématiques, Université de Reims, Champagne-Ardennes region scholarship, on blood flow simulations in brain vascular networks – due end of 2015;

5. Imen Melki, co-supervised with Laurent Najman and Jean Cousty, LIGM, ESIEE, on a CIFRE scholarship with General Electric Healthcare – due end of 2014 ;

- Past students, now all doctors :

6. Phuc NGO, co-supervised with Michel Couprie and Yukiko Kenmochi, A3SI, research ministry scholarship, on discrete transformations for optimal registration – graduated 18th October 2013, now post-doc at CEA;
7. Ania Jezierska, co-supervised with Jean-Christophe Pesquet and Caroline Chaux, Univ.-Marne-la-Vallée, ANR DIAMOND 2009 scholarship, on image restoration for confocal macroscopy, graduated 13th May 2013, now post-doc at Université Paris-Est;
8. Nicolas Combaret, co-supervised with Dominique Bernard, CNRS. Aquitaine Region scholarship, on the study of materials in X-ray microtomography – graduated 12th December 2012, now research engineer at FEI vizualisation science group (`http://www.vsg3d.com`).
9. Lásló Marak, co-supervised with Laurent Najman, A3SI, research ministry scholarship, on fast and globally optimal segmentation methods for 3D data - defended on 28th March 2012, now research engineer at Real Eyes Inc, Budapest ;
10. Camille Couprie, co-supervised avec Laurent Najman, A3SI, and Leo Grady, Siemens Corporate Research, Princeton. Defense ministry (DGA) scholarship 2008, on optimal discrete and continuous filtering and segmentation methods, applied to motion segmentation and biometry – defended on 10th Octobre 2011. Camille is now a research engineer at IFP-EN (Institut Français du Pétrole-Energies Nouvelles). Camille's thesis received three prestigious awards: the industrial AEDS best multisciplinary thesis 2012, the Société Informatique de France accessit (2nd prize) for the Gilles Kahn award 2012, and the defence ministry DGA prize 2013 ;
11. Olena Tankyevych, co-supervised with Gilles Bertrand, A3SI et Petr Dokladal, ENSMP. CNRS-DGI 2007 scholarship, on the segmentation of thin objects in 3D applied to medical imaging – defended on 19th Octobre 2010, now a Maître de Conférence at Université Paris-Est Créteil;
12. Harold Phélippeau, co-supervised with Mohamed Akil, A3SI. CIFRE scholarship with Philips and NxP France, on the improvement of image quality for digital photography – defended on 3rd April 2009, now research engineer at Noesys France.
13. Benjamin Appleton, Australian education and CSIRO joint scholarship, on segmentation methods by globally optimal methods, received 2005, now research engineer at Google Australia.
14. Fiona Evans, Australian education and CSIRO joint scholarship, on segmentation methods using EM-type statistical methods, received 2005, now research engineer in the public sector, Perth, Australia.

- Other supervisions : many 3rd to 5th university year students, typically 3-6 months, between 1995 and present. Apology for not stating all the names here.

Teaching

Duties at ESIEE represent about 300h of actual teaching a year, spit between lecturing, tutorials and practical sessions. Specific duties include:

- Member of the initial and core group of academics responsible for creating and running the Institut Supérieur des Bio-Sciences (ISBS), since its debut in 2004. This institute delivers an engineering diploma after 3 years of study, corresponding to the 3rd to 5th year of university. The institute is habilitated by the French engineering title commission. The diploma is delivered by the Université Paris-Est Créteil. 24 students graduate each year. In particular, I supervise the last year of study, particularly the imaging major, internships, and the numerical sciences courses.
- Courses : at ESIEE and ISBS I teach mathematical morphology, image analysis, mathematical optimization, programming, operating systems, graph theory, computer languages and compiling.
- Many academic supervision of internships and applied projects.

Computer science skills

- Languages : C,C++,S/Splus/R,Fortran,Python,lex+yacc,LaTeX, various scripts.
- Environments : Linux/Unix, Windows, MacOS, RTOS.
- Expertise in : operating systems, GUIs, real-time and parallel programming.

R&D projects

This list is non-exhaustive.

- Between 1990-1996, project "mesure of man-made insulation fibres", with Isover Saint-Gobain. Research engineering with co-supervision of 4 technicians.
- Between 1994-2004, project "automated image-based diagnosis of skin cancer" with SME Polartechnics Limited (Australia). R&D and principal research duties in imaging. Liaison between engineers and medical doctors. Budget of about AU$ 40k/year. The project resulted in a monitoring and diagnosis medical instrument that received several prizes and patents, the Solarscan, which was sold worldwide.
- Between 2000-2004, project "development of an automated fluorescence microscope for high-content screening and cellomics", with SME Axon Limited (USA). Principal architect for the image analysis library used in the resulting instrument: the imageXpress 5000, which was a technical and financial success. Line manager for two engineers. Budget of about AU$ 80k/an. This project ultimately resulted in a significant valuation for Axon Instruments, prior to acquisition by Molecular Devices in 2004. The instrument remained unchanged on MD's catalogue until 2010. See appendix C for some details.
- Between 2005-present. Contributions to the following ANR (Agence Nationale de la Recherche) projects, including proposal writing, academic, research and administrative duties:

1. SURF 2005 (as principal investigator) with ENPC, CEREMADE, ECP et ESIEE, computation of continuous and discrete minimal surfaces, 400k€;
2. MICROFISS 2007 with CNRS, Université de Lille et ESIEE, study of concrete aging via X-Ray microtomography, 300k€;
3. DIAMOND 2009 with INRA, INRIA, Université de Haute Alsace, Institut Pasteur et Institut Gaspard Monge: image restoration for confocal MACROscopy, 650k€;
4. KIDICO 2010 (ESIEE head investigator) avec Université de Strasbourg et Université de Clermond-Ferrand: discrete geometry, morphology and discrete calculus, 350k€.
5. VIVABRAIN 2012 (ESIEE head investigator) with Université de Strasbourg, Institut National Polytechnique de Grenoble, Université de Reims, and Kitware, Inc: segmentation and blood flow simulation in cerebral blood vessels. 980k€.

- Initiating, conducting and managing small industrial project too numerous to be listed here (5-30k€), with Powercor Australia, Atlab Australia, Sanofi France, EDF, Lafarge, Noveltis, Emphron Australia...

Appendix B

Publications list

This part contains my list of publications

Publications Hugues Talbot

13 mai 2013

Books and special editions

[1] H. Talbot and R. Beare, editors. *Mathematical Morphology, Proceedings of the 6th international symposium*. CSIRO Publishing, Sydney, Australia, April 2002. ISBN 978-0643068049.

[2] C. Sun, H. Talbot, S. Ourselin and T. Adriaansen. *Digital image computing, techniques and applications, Proceedings of the VIIth Biennial Australian Pattern Recognition Society Conference*. 2003. 2 volumes, ISBN 0-643-09040-X.

[3] B. Lamiroy, L. Najman and H. Talbot. *Systèmes d'exploitation*. Pearson Education, 2006. ISBN 978-2744071935.

[4] L. Najman and H. Talbot, editors. *Morphologie mathématique 1 : approches déterministes*. Hermes, Paris, 2008. ISBN 978-2746218413.

[5] L. Najman and H. Talbot, editors. *Mathematical Morphology : from theory to applications*. ISTE-Wiley, London, UK, September 2010. ISBN 978-1848212152.

[6] H. Talbot and L. Najman, editors. *Morphologie mathématique 2 : estimation, choix et mise en œuvre*. Traité Signal et Image. Hermes, September 2010. ISBN 978-2746225930.

Book chapters

[7] G. Day and H. Talbot. Automated Melanoma Diagnosis. In R. Wootton and A. Oakley, editors, *Teledermatology*, chapitre 22, pages 275–286. The Royal Society of Medicine Press, Ltd., 2002. ISBN 1-85315-507-1.

[8] L. Najman and H. Talbot. Introduction à la morphologie mathématique. In *Morphologie mathématique 1 : approches déterministes*, chapitre 1, pages 19–45. Lavoisier, 2008. ISBN 978-2746218413.

[9] T. Géraud, H. Talbot and M. Vandroogenbroeck. Morphologie et Algorithmes. In Talbot and Najman [6], chapitre 6, pages 151–180. ISBN 978-2746225930.

[10] M. Couprie and H. Talbot. Distances, granulometries and skeletons. In Najman and Talbot [5], chapitre 10, pages 265–291. ISBN 978-1848212152.

[11] M. Couprie and H. Talbot. Distances, granulométries, squelettes. In Talbot and Najman [6], chapitre 4, pages 93–118. ISBN 978-2746225930.

[12] T. Géraud, H. Talbot and M. Vandroogenbroeck. Algorithms for mathematical morphology. In Najman and Talbot [5], chapitre 12, pages 323–354. ISBN 978-1848212152.

[13] L. Najman and H. Talbot. Introduction to mathematical morphology. In *Mathematical Morphology : from theory to applications* [5], chapitre 1, pages 3–34. ISBN 978-1848212152.

[14] H. Talbot, J. Serra and L. Najman. Introduction to measurement theory for image analysis. In Najman and Talbot [5], chapitre 4, pages 111–132. ISBN 978-1848212152.

[15] H. Talbot, J. Serra and L. Najman. Introduction à la mesure en analyse d'images. In Talbot and Najman [6], chapitre 1, pages 23–44. ISBN 978-2746225930.

[16] A. Nakib, L. Najman, H. Talbot and P. Siarry. Application du partitionnement de graphes à la segmentation d'images. In C.-E. Bichot and P. Siarry, editors, *Partitionnement de graphe : optimisation et applications*, Traité IC2, chapitre 11, pages 319–344. Hermes-Lavoisier, 2011. ISBN 978-2746230057.

[17] C. Couprie, L. Najman and H. Talbot. Seeded Segmentation Methods for Medical Image Analysis. In G. Dougherty, editor, *Medical Image Processing : Techniques and Applications*, pages 27–58. Springer, 2011. ISBN 978-1441997692.

[18] O. Tankyevych, H. Talbot, N. Passat, M. Musacchio and M. Lagneau. Angiographic Image Analysis. In G. Dougherty, editor, *Medical Image Processing : Techniques and Applications*, pages 115–145. Springer, 2011. ISBN 978-1441997692.

Journal Articles

[19] H. Talbot. Le prétraitement en morphologie mathématique. *Journal de Métallurgie et Matériaux*, 1993.

[20] H. Talbot and D. Jeulin. Estimation of Fibre Length and Diameter Distribution from SEM Images. *Acta Stereologica*, 13-2 :357–362, 1993.

[21] H. Talbot, D. Jeulin and D. Hanton. Image analysis of insulation mineral fibres. *Microscopy, microanalysis and microstructures*, 7 :361–368, October-December 1996.

[22] M. Vandroogenbroeck and H. Talbot. Fast Computation of Morphological Operations with Arbitrary Structuring Elements. *Pattern Recognition Letters*, 17 :1451–1460, 1996.

[23] E. Breen, R. Jones and H. Talbot. The Morphological Approach to Industrial Image Analysis Applications. *Acta Stereologica*, 16(3) :233–240, 1997.

[24] E. Breen, R. Jones and H. Talbot. Mathematical Morphology, a Useful Set of Tools for Image Analysis. *Statistics and Computing*, 10 :105–120, 2000.

[25] H. Talbot, T. Lee, D. Jeulin, D. Hanton and L. W. Hobbs. Image Analysis of Insulation Mineral Fibres. *Journal of Microscopy*, 200(3) :251–268, Dec 2000.

[26] P. Soille and H. Talbot. Directional Morphological Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11) :1313–1329, 2001.

[27] L. Bischof, M. Buckley, R. Lagerstrom, C. Sun, H. Talbot, D. Wang and P. Vallotton. Image analysis of neurite branching : High-content screening at high speed. *AMERICAN BIOTECHNOLOGY LABORATORY*, 23(10) :22, 2005.

[28] B. Appleton and H. Talbot. Recursive Filtering of Images with Symmetric Extension. *Signal Processing*, 85 :1546–1556, 2005.

[29] B. Appleton and H. Talbot. Globally optimal geodesic active contours. *Journal of Mathematical Imaging and Vision*, (23) :67–86, 2005.

[30] H. Heijmans, M. Buckley and H. Talbot. Path openings and closings. *Journal of Mathematical Imaging and Vision*, 22 :107–119, 2005.

[31] S. Menzies, L. Bischof, H. Talbot, A. Gutenev, M. Avramidis, L. Wong, S. K. Lo, G. Mackellar, V. Skladnev, W. McCarthy, J. Kelly, B. Cranney, P. Lye, H. Rabinovitz, M. Oliviero, A. Blum, A. Virol, B. DeAmbrosis, R. McCleod, H. Koga, C. Grin, R. Braun and R. Johr. The Performance of SolarScan : An Automated Dermoscopy Image Analysis Instrument for the Diagnosis of Primary Melanoma. *Archives of Dermatology*, 141(11) :1388–1396, November 2005.

[32] B. Appleton and H. Talbot. Globally Minimal Surfaces by Continuous Maximal Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1) :106–118, 2006.

[33] C. Sun, R. Jones, H. Talbot, X. Wu, K. Cheong, M. Beare, M. Buckley and M. Berman. Measuring the Distance of Vegetation from Powerlines Using Stereo Vision. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4) :269–283, June 2006.

[34] H. Talbot and B. Appleton. Efficient complete and incomplete paths openings and closings. *Image and Vision Computing*, 25(4) :416–425, April 2007.

[35] O. Le Bihan, P. Bonnafous, L. Marak, S. Trépout, S. Mornet, H. Talbot, J.-C. Taveau and O. Lambert. Cryo-electron tomography of nanoparticle transmigration into liposome. *Journal of Structural Biology*, 168 :419 – 425, 2009.

[36] C. Chaux, A. Jezierska, J.-C. Pesquet and H. Talbot. A Spatial Regularization Approach for Vector Quantization. *Journal of Mathematical Imaging and Vision*, pages 1–16, 2010. 10.1007/s10851-010-0241-3.

[37] J. Chaussard, M. Couprie and H. Talbot. Robust skeletonization using the discrete lambda-medial axis. *Pattern Recognition Letters*, 32(9) :1384–1394, July 2010.

[38] S. Valero, J. Chanussot, J. Benediktsson, H. Talbot and B. Waske. Advanced Directional mathematical Morphology for the Detection of the Road Network in very high resolution images. *Pattern Recognition Letters*, 31(10) :1120–1127, July 2010.

[39] F. Cazettes, J. I. Cohen, P. L. Yau, H. Talbot and A. Convit. Obesity-mediated inflammation may damage the brain circuit that regulates food intake. *Brain Research*, 1373(0) :101 – 109, 2011.

[40] C. Couprie, L. Grady, L. Najman and H. Talbot. Power watersheds : A Unifying Graph-Based Optimization Framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7) :1384 – 1399, 2011.

[41] C. Couprie, L. Grady, H. Talbot and L. Najman. Combinatorial Continuous Maximum Flows. *SIAM Journal on Imaging Sciences*, 4(3) :905–930, 2011.

[42] L. Marak, O. Tankyevych and H. Talbot. Continuous maximum flow segmentation method for nanoparticle interaction analysis. *Journal of Microscopy*, 244(1) :59–78, October 2011.

[43] R. Zrour, Y. Kenmochi, H. Talbot, L. Buzer, Y. Hamam, I. Shimizu and A. Sugimoto. Optimal consensus set for digital line and plane fitting. *International Journal of Imaging Systems and Technology*, 21 :45–57, 2011.

[44] P. Ngo, Y. Kenmochi, N. Passat and H. Talbot. Combinatorial structure of rigid transformations in 2D digital images. *Computer Vision and Image Understanding*, November 2012. In print.

[45] A. Dufour, O. Tankyevych, B. Naegel, H. Talbot, C. Ronse, J. Baruthio, P. Dokladal and N. Passat. Filtering and segmentation of 3D angiographic data : Advances based on mathematical morphology. *Medical Image Analysis*, (0) :–, 2012. In print.

[46] F. Cokelaer, H. Talbot and J. Chanussot. Efficient Robust d-Dimensional Path Operators. *IEEE Journal of Selected Topics in Signal Processing*, 6(7) :830 –839, nov. 2012.

[47] E. Chouzenoux, A. Jezierska, J.-C. Pesquet and H. Talbot. A Majorize-Minimize Subspace Approach for \ell_2-\ell_0 Image Regularization. *SIAM Journal on Imaging Sciences*, 6(1) :563–591, 2013.

[48] E. Már Sigurdsson, S. Valero-Valbuena, J. A. Benediktsson, J. Chanussot, H. Talbot and E. Stefansson. Automatic retinal vessel extraction based on directional mathematical morphology and fuzzy classication. *in preparation*, 2013.

[49] C. Couprie, L. Grady, L. Najman, J.-C. Pesquet and H. Talbot. Dual constrained TV-Based regularization on graphs. *SIAM Journal on Imaging Sciences*, 2013. In print.

Refereed conferences articles

[50] H. Talbot, D. Jeulin and L. W. Hobbs. Scanning Electron Microscope Image Analysis of Fiber Glass Insulation. In *Proc. 50th Annual Meeting of the Electron Microscopy Society of America*, pages 994–995, Boston, 1992. San Fransisco Press.

[51] H. Talbot and I. Terol Villalobos. Binary image segmentation using weighted skeletons. In *Image algebra and morphological image processing III*, volume 1769, pages 147–155, San Diego, CA, SPIE, July 1992.

[52] H. Talbot and L. Vincent. Euclidean skeleton and Conditional Bisectors. In *Visual Communications and Image Processing'92*, volume 1818, pages 862–873, Boston, SPIE, November 1992.

[53] C. Gratin, F. Meyer and H. Talbot. Fast Gray-Level Morphological Transforms with Any Structuring Elements. In *proceedings of SPIE conference on Visual Communications and Image Processing*, Boston, November 1993.

[54] C. Vachier, F. Meyer, C. Gratin and H. Talbot. Fitrage par décomposition morphologique : application à l'extraction de structures rectilignes. In *Proceedings RFIA*, volume 94, pages 255–263. INRIA, 1994.

[55] T. Lee and H. Talbot. A Fast Method for Detecting and Matching Linear Features in Images. In *Proceedings of Digital Image Computing, Techniques and Applications (DICTA)*, pages 649–654, Brisbane, Australia, Dec. 1995. Australian Pattern Recognition Society.

[56] H. Talbot. A Morphological Algorithm for Linear Segment Detection. In P. Maragos, R. W. Schafer and M. A. Butt, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 219–226, Atlanta, GA, USA, May 1996. proceedings for the third International Symposium on Mathematical Morphology, Kluwer.

[57] T. C. M. Lee and H. Talbot. Automatic Reconnection of Linear Segments by the Minimum Description Length Principle. In *Digital Image Computing : Techniques and Applications*, pages 555–560, Dec. 1997.

[58] L. Bischof, H. Talbot, E. Breen, D. Lovell, D. Chan, G. Stone, S. Menzies, A. Gutenev and R. Caffin. An Automated Melanoma Diagnosis System. Ballarat, Vic., Australia, 1998.

[59] P. Soille and H. Talbot. Image structure orientation using mathematical morphology. volume 2, pages 1467–1469, Brisbane, August 1998. ICPR, IEEE Computer Society.

[60] H. Talbot, C. Evans and R. Jones. Complete Ordering and Multivariate Mathematical Morphology. In H. Heijmans and J. Roerdink, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 11 of *Computational Imaging and Vision*, pages 27–34, Amsterdam, June 1998. International Society for Mathematical Morphology, Kluwer. Proceedings of the fourth international symposium on mathematical morphology.

[61] R. Beare and H. Talbot. Exact Seeded Region Growing for Image Segmentation. In *Proceedings of Digital Image Computing, Techniques and Applications (DICTA)*, Perth, WA, Australia, December 1999. Australian Pattern Recognition Society.

[62] L. Bischof, H. Talbot, E. Breen, D. Lovell, D. Chan, G. Stone, S. Menzies, A. Gutenev and R. Caffin. An automated melanoma diagnosis system. In B. Pham, M. Braun, A. J. Maeder and M. P. Eckert, editors, *New Approaches in Medical Image Analysis*, volume 3747, pages 130–141. SPIE, 1999.

[63] Z. Lin, J. Jin and H. Talbot. Unseeded region growing for 3D image segmentation. In *Selected papers from the Pan-Sydney workshop on Visualisation-Volume 2*, pages 31–37. Australian Computer Society, Inc., 2000.

[64] R. Beare and H. Talbot. Motion segmentation using seeded region growing. In L. Vincent and D. Bloomberg, editors, *Mathematical morphology and its application to Image analysis, proceedings of the 5th International Symposium on Mathematical Morphology (ISMM)*, Palo Alto, California, June 2000. Kluwer.

[65] M. Buckley and H. Talbot. Flexible linear openings and closings. In L. Vincent and D. Bloomberg, editors, *Mathematical Morphology and its application to image analysis*, pages 109–118, Palo Alto, June 2000. ISMM, Kluwer.

[66] H. Talbot and B. Appleton. Elliptical distance transforms and the object splitting problem. In H. Talbot and R. Beare, editors, *Mathematical Morphology, Proceedings of the 6th international symposium*, pages 229–240, Sydney, Australia, April 2002. CSIRO Publishing.

[67] B. Appleton and H. Talbot. Efficient and Consistent Recursive Filtering of Images with Reflective Extension. In L. Griffin and M. Lillholm, editors, *Proceedings of the IVth International Conference on Scale-Space theories in Computer Vision*, pages 699–712. British Machine Vision Association, Springer LNCS-2695, June 2003.

[68] H. Talbot and L. Bischof. An Overview of the Polartechnics Solarscan Melanoma Diagnosis Algorithms. In B. C. Lovell and A. J. Maeder, editors, *WDIC 2003*, pages 33–38, U. of Queensland, St. Lucia, Brisbane, February 2003. APRS.

[69] H. Heijmans, M. Buckley and H. Talbot. Path-based morphological openings. In *Proceedings of IEEE ICIP 2004*, pages 3085–3088, Singapore, October 2004.

[70] B. Appleton and H. Talbot. Efficient path openings and closings. In C. Ronse, L. Najman and E. Decencière, editors, *Mathematical Morphology : 40 Years On*, volume 30 of *Computational Imaging and Vision*, pages 33–42, Dordrecht, 2005. Springer-Verlag.

[71] H. Talbot. Elliptical Distance Transforms and Applications. In *Proceedings of the 13th IAPR conference on Discrete Geometry for Computer Imagery (DGCI)*, pages 320–330, Szeged, Hungary, October 25-27 2006.

[72] N. Combaret and H. Talbot. Robust 3D segmentation of composite materials fibres. In G. J. F. Banon, J. Barrera, U. d. M. Braga-Neto and N. S. T. Hirata, editors, *Proceedings of ISMM 2007*, volume 2, pages 25–26, São José dos Campos, October 10–13 2007. Universidade de São Paulo (USP), Instituto Nacional de Pesquisas Espaciais (INPE).

[73] H.Phelippeau, M.Akil, T.Fraga, S.Bara and H.Talbot. Demosaicing on Trimedia 3270. In *International Workshop on Design and Architectures for Signal and Image Processing*, 2007.

[74] O. Tankyevych, L. Marak, H. Talbot and P. Dokladal. Segmentation of 3D nano-scale polystyrene balls. In G. J. F. Banon, J. Barrera, U. d. M. Braga-Neto and N. S. T. Hirata, editors, *International Symposium on Mathematical Morphology - International Symposium on Mathematical Morphology'07, 8th International Symposium, Proceedings*, volume 2, pages 37–38. INPE, 2007.

[75] C. Torres, P. Perrot and H. Talbot. Face recognition : from biometrics to forensic applications. In *conference on Biometrical Feature Identification and Analysis (BFIA)*, Göttingen, September 2007. URL : http ://www.stochastik.math.uni-goettingen.de/biometrics2007/talks/Torres.pdf.

[76] O. Tankyevych, H. Talbot and P. Dokládal. Curvilinear morpho-Hessian filter. In *proceedings of the International Symposium on Biomedical Imaging (ISBI)*, pages 1011–1014, 2008.

[77] G. Lobo-Pappa, D. Menotti, H. Talbot and M. Meignan. Towards Automated Lymphoma Prognosis based on PET Images. pages 279–284, Cancún, Mexico, October 2008. International IEEE Workshop on Machine Learning for Signal Processing (MLSP).

[78] L. Marak, H. Talbot, O. Lambert and J.-C. Taveau. Segmentation techniques for the analysis of electron nano-tomography images. In *3D-IMS*, Carcans-Maubuisson, September 2008.

[79] L. Najman, J. Cousty, M. Couprie, H. Talbot, S. Clément-Guinaudeau, T. Goissen and J. Garot. An open, clinically-validated database of 3D+t cine-MR images of the left ventricle with associated manual and automated segmentation. In *ISC/NA-MIC Workshop on Open Science at MICCAI 2007*, number 157. The Insight Journal, 2008. url=`http://hdl.handle.net/1926/550`.

[80] H. Phelippeau, H. Talbot, S. Bara and M. Akil. Shot-noise adaptive bilateral filter. pages 4–8, Beijing, China, October 2008. International Conference on Signal Processing (ISCP), IEEE.

[81] M. Poilpre, P. Perrot and H. Talbot. Image Tampering Detection Using Bayer Interpolation and JPEG Compression. In *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia (eForensics'2008)*, number 17, page 5. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008. ISBN :978-963-9799-19-6.

[82] H. Talbot, B. Appleton and L. Marak. Continuous Maximal Flows and Applications. In *Graph Cuts and Related Discrete or Continuous Optimization Problems*, Los Angeles, February 2008. IPAM, UCLA. url=`https://www.ipam.ucla.edu/schedule.aspx?pc=gc2008`.

[83] J. Chaussard, M. Couprie and H. Talbot. A discrete lambda-medial axis. In S. Brlek, C. Renautauer and X. Provençal, editors, *Proceedings of the 15h IAPR conference on Discrete Geometry for Computer Imagery (DGCI)*, volume 5810 of *Lecture Notes in Computer Science*, pages 421–433, Montreal, Canada, September 2009. Springer.

[84] C. Couprie, L. Grady, L. Najman and H. Talbot. Power watersheds : A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *Proceedings of ICCV 2009*, pages 731–738, Kyoto, Japan, 2009. IEEE.

[85] L. Marak, J. Cousty, L. Najman and H. Talbot. 4D Morphological segmentation and the miccai LV-segmentation grand challenge. In *MICCAI 2009 Workshop on Cardiac MR Left Ventricle Segmentation Challenge*, MIDAS Journal, pages 1–8. MIDAS, November 2009.

[86] H. Phelippeau, M. Akil, H. Talbot and S. Bara. Algorithme de Dématriçage pour la Photographie Numérique : Architecture Dédiée et Implantation sur FPGA. In *Proceedings of GRETSI*, Dijon, September 2009.

[87] H. Talbot, H. Phelippeau, M. Akil and S. Bara. Efficient Poisson Denoising for Photography. In *Proceedings of ICIP 2009*, pages 3881–3883, Cairo, Egypt, 2009.

[88] O. Tankyevych, H. Talbot, P. Dokladal and N. Passat. Direction-Adaptive Grey-level Morphology. Application to 3D Vascular Brain Imaging. In *Proceedings of ICIP 2009*, pages 2261–2264, Cairo, Egypt, 2009. IEEE.

[89] O. Tankyevych, H. Talbot, P. Dokladal and N. Passat. Spatially-variant morpho-Hessian filter : efficient implementation and application. In *Mathematical Morphology and Its Application to Signal and Image Processing, Proceedings of the 9th International Symposium on Mathematical Morphology (ISMM) 2009*, pages 137–148, Groningen, The Netherlands, 2009.

[90] S. Valero, J. Chanussot, J. Benediktsson and H. Talbot. Détection automatique du réseau vasculaire rétinien basée sur la morphologie directionnelle et la fusion de décision. In *Proceedings of GRETSI*, Dijon, France, 2009. INIST-CNRS. Paru.

[91] S. Valero, J. Chanussot, J. Benediktsson, H. Talbot and B. Waske. Directional Mathematical Morphology For the detection of the road network in very high resolution remote sensing images. In *Proceedings of ICIP 2009*, pages 3725–3728, Cairo, Egypt, 2009.

[92] R. Zrour, Y. Kenmochi, H. Talbot, L. Buzer, Y. Hamam, I. Shimizu and A. Sugimoto. Optimal consensus set for digital plane fitting. In *Proceedings of 3DIM 2009, workshop of ICCV*, Kyoto, Japan, 2009.

[93] R. Zrour, Y. Kenmochi, H. Talbot, L. Buzer, Y. Hamam, I. Shimizu and A. Sugimoto. Optimal consensus set for digital line fitting. In *Proceedings of IWCIA 2009*, Cancun, Mexico, 2009.

[94] R. Zrour, Y. Kenmochi, H. Talbot, I. Shimizu and A. Sugimoto. Combinatorial optimization for fitting of digital line and plane. In *Proceedings of PSIVT'09*. CVIM Workshop, january 2009.

[95] D. Aiger and H. Talbot. The Phase Only Transform for unsupervised surface defect detection. In *Computer Vision and Pattern Recognition (CVPR 2010)*, pages 295–302, San Francisco, CA, June 2010. IEEE.

[96] C. Couprie, L. Grady, H. Talbot and L. Najman. Anisotropic diffusion using power watersheds. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 4153–4156, Honk-Kong, September 2010.

[97] A. Jezierska, C. Chaux, H. Talbot and J.-C. Pesquet. image quantisation under spatial smoothness constraints. In *Proceedings of ICIP 2010*, Honk-Kong, September 2010. IEEE signal processing society.

[98] Y. Kenmochi, L. Buzer and H. Talbot. Efficiently Computing Optimal Consensus of Digital Line Fitting. In *20th International Conference on Pattern Recognition, ICPR 2010*, pages 1064–1067, Istambul, Turkey, August 2010.

[99] A. Kornev, L. Babout, M. Janaszewski and H. Talbot. Outer Surface Reconstruction for 3D Fractured Objects. In L. Bolc, R. Tadeusiewicz, L. J. Chmielewski and K. W. Wojciechowski, editors, *International Conference on Computer Vision and Graphics (ICCVG)*, volume 6375 of *Lecture Notes in Computer Science*, pages 57–64. Springer, 2010.

[100] D. Aiger, Y. Kenmochi, H. Talbot and L. Buzer. Efficient robust digital hyperplane fitting with bounded error. In *Proceedings of the 16th IAPR Conference on Discrete Geometry and Computer Imagery (DGCI)*, pages 223–234, 2011.

[101] E. Chouzenoux, J. Pesquet, H. Talbot and A. Jezierska. A memory gradient algorithm for l2 - l0 regularization with applications to image restoration. In *International Conference on Image Processing (ICIP)*, Brussels, Belgium, 11-14 September 2011.

[102] C. Couprie, X. Bresson, L. Najman, H. Talbot and L. Grady. Surface reconstruction using Power Watershed. In *International Symposium on Mathematical Morphology (ISMM)*, Lake Maggiore, Italy, July 6-8th 2011.

[103] C. Couprie, H. Talbot, J. Pesquet, L. Najman and L. Grady. Dual constrained TV-based regularization. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 22-27 2011.

[104] A. Jezierska, C. Chaux, J. Pesquet and H. Talbot. An EM approach for Poisson-Gaussian noise modeling. In *European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, 29 August - 2 September 2011.

[105] A. Jezierska, H. Talbot, O. Veksler and D. Wesierski. A fast solver for truncated-convex priors : quantized-convex split moves. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, Saint Petersburg, Russia, 25-27 July 2011.

[106] I. Melki, H. Talbot, J. Cousty, L. Najman, C. Pruvot, J. Knoplioch and L. Launay. A hybrid algorithm for automatic heart delineation in CT angiography. In *Computer Assisted Radiology and Surgery (CARS)*, France, 2012. hal-00730481.

[107] V. Bismuth, R. Vaillant, H. Talbot and L. Najman. Curvilinear Structure Enhancement with the Polygonal Path Image - Application to Guide-Wire Segmentation in X-Ray Fluoroscopy. In N. Ayache, H. Delingette, P. Golland and K. Mori, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, volume 7511 of *Lecture Notes in Computer Science*, pages 9–16, Nice, France, September 2012. Springer Berlin Heidelberg.

[108] P. Ngo, Y. Kenmochi, N. Passat and H. Talbot. Combinatorial Properties of 2D Discrete Rigid Transformations under Pixel-invariance Constraints. In *Proceedings of the 15h International Workshop on Combinatorial Image Analysis (IWCIA)*, Austin, TX, November 2012.

[109] A. Jezierska, H. Talbot, C. Chaux, J. Pesquet and G. Engler. Poisson-Gaussian noise parameter estimation in fluorescence microscopy imaging. In *9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1663 –1666, may 2012.

[110] A. Jezierska, E. Chouzenoux, J.-C. Pesquet and H. Talbot. A primal-dual proximal splitting approach for restoring data corrupted with poisson-gaussian noise. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1085 –1088, march 2012.

[111] I. Melki, H. Talbot, J. Cousty, C. Pruvot, J. Knoplioch, L. Launay and L. Najman. A hybrid algorithm for automatic heart segmentation in CT angiography. In *IEEE 19th International Conference on Image Processing-ICIP'2012*, September 2012.

[112] A. Dufour, C. Ronse, J. Baruthio, O. Tankyevych, H. Talbot and N. Passat. Morphology-Based Cerebrovascular Atlas. In *Proceedings of ISBI*, San Francisco, CA, 2013.

[113] E. Chouzenoux, F. Zolyniak, E. Gouillart and H. Talbot. A Majorize-Minimize Memory Gradient algorithm applied to X-ray tomography. In *Proceedings of ICIP*, Melbourne, Australia, 2013.

[114] N. Phuc, N. Passat, Y. Kenmochi and H. Talbot. Well-composed images and rigid transformations. In *Proceedings of ICIP*, Melbourne, Australia, 2013.

[115] M. Phan, Y. Kenmochi, A. Sugimoto, H. Talbot, E. Andres and R. Zrour. Efficient Robust Digital Annulus Fitting with Bounded Error. In R. Gonzalez-Diaz, M.-J. Jimenez and B. Medrano, editors, *Discrete Geometry for Computer Imagery*, volume 7749 of *Lecture Notes in Computer Science*, pages 253–264. Springer, 2013.

[116] O. Tankyevych, H. Talbot and N. Passat. Semi-connections and hierarchies. In *Proceedings of ISMM*, Uppsala, Sweden, 2013.

[117] P. Ngo, Y. Kenmochi, N. Passat and H. Talbot. Sufficient conditions for topological invariance of 2D images under rigid transformations. In *Proceedings of DGCI*, Seville, Spain, 2013.

[118] C. Jaquet, E. Ando, G. Viggiani and H. Talbot. Estimation of separating planes between touching 3D objects using power watershed. In *Proceedings of ISMM*, Uppsala, Sweden, 2013.

[119] T.-A. Nguyen, A. Dufour, O. Tankyevych, A. Nakib, E. Petit, H. Talbot and N. Passat. This structure filtering framework with non-local means, Gaussian derivatives and spatially-variant mathematical morphology. In *International Conference in Image Processing (ICIP)*, Melbourne, Australia, September 2013.

Theses

[120] H. Talbot. Mesures de diamètres de fibres de verre d'isolation par analyse automatique d'image. Rapport de maitrise, Université Paris VI, 1990.

[121] H. Talbot. *Analyse morphologique de fibres minérales d'isolation.* Thèse de doctorat, Ecole des Mines de Paris, October 1993.

Technical reports

[122] H. Talbot and T. Lee. Automated Image Analysis of Insulation Mineral Fibres, First Stage Review Report. Technical Report DMS — E 95/48, CSIRO, 1995.

[123] L. Bischof, E. J. Breen, H. Talbot and S. Menzies. Melanoma Project Report - End of Phase II. Technical Report CMIS — 98/68, CSIRO-MIS, 1998.

[124] H. Talbot, M. Buckley, M. Berman, R. Lagerstrom and K. Cheong. Axon CellPix Project : Task B report. Technical Report 01/157, CMIS, 2001.

[125] H. Talbot, M. Buckley, M. Berman, R. Lagerstrom and K. Cheong. Axon CellPix Project : Task B report, Appendix. Technical Report 01/157, CMIS, 2001.

[126] H. Talbot, M. Buckley, M. Berman, R. Lagerstrom and K. Cheong. Axon ImageXPress Project : Task F intermediate report, Offline Cell Nuclei Segmentation. Technical Report 02/xx1, CMIS, 2002.

[127] C. Couprie, L. Grady, H. Talbot and L. Najman. DCMF : Discretized Continuous Max Flows. Technical Report IGM 2008-07, Institut Gaspard Monge, CNRS UMR 8049, 2008. collaboration with Siemens USA.

[128] H. T. Rita Zrour, Yukiko Kenmochi, L. Buzer, Y. Haman, I. Shimizu and A. Sugimoto. Optimal consensus set for digital line and plane fitting. Technical Report IGM 2009-06, Institut Gaspard Monge, CNRS UMR 8049, 2009.

Patents

[129] C. Sun, H. Talbot, K. Cheong, X. Wu, M. Buckley, R. Beare, R. Jones and M. Berman. Corridor mapping system and method, April 2002. WO 2002/082181.

[130] V. Skladnev, A. Gutenev, S. Menzies, L. Bischof, H. Talbot, E. Breen and M. Buckley. Diagnostic Feature Extraction in Dermatological Examination, August 2004. WO 02/094098 A1 and US 2004/0267102 A1.

[131] V. Skladnev, S. Menzies, A. Gutenev, A. Baltrac, D. Varvel, L. Bischof and H. Talbot. Boundary Finding in Dermatological Examination, december 2004. US 2004/0264749 A1.

[132] V. Skladnev, A. Gutenev, S. Menzies, R. Thompson, A. Baltrac, L. Bischof, R. Caffin, S. Rowe, H. Talbot, D. Varvel and P. West. System and method for examining, recording and analyzing dermatological conditions, january 2006. US Patent 6993167 B1.

[133] H.Phelippeau, H.Talbot, S.Bara and M.Akil. Image filtering, 2008. European Patent EP 2 223 284 B1.

[134] H. Phelippeau, S. Bara, H. Talbot and M. Akil. Image filtering, June 2012. US Patent 8199215.

Appendix C

Industrial contributions

This part contains some relevant industrial contributions.

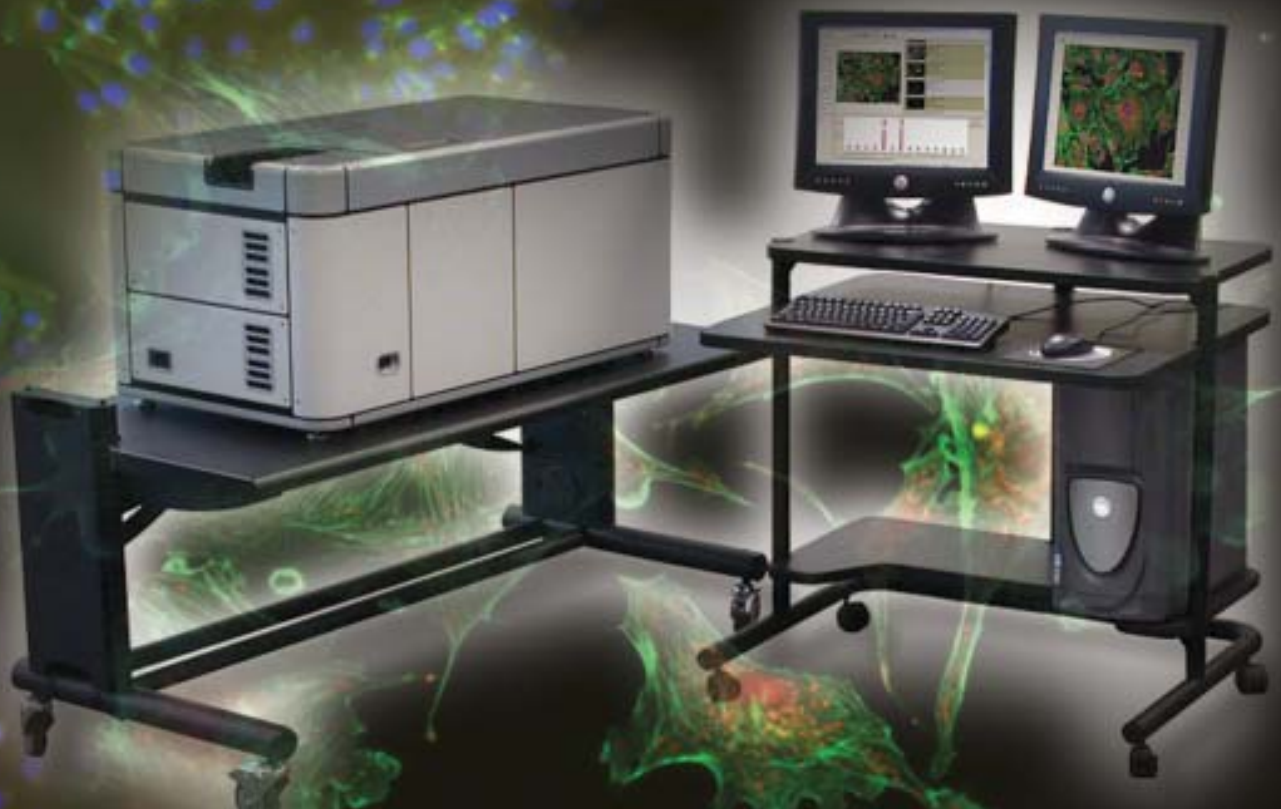
C.1 Axon Instruments imageXpress 5000

The first is a partial brochure of the instrument ImageXpress 5000, which I helped develop between 2000 and 2004. I was in charge of the entire image analysis library used in the context of this project. All the assays depended on it. This instrument was sold between 2003 and 2010. The full brochure is at `http://htbc.stanford.edu/equipment/ImageXpress_Brochure.pdf`.

IMAGEXPRESS

Automated Cellular Imaging and Analysis System

- *CELLULAR IMAGING AND ANALYSIS FOR SCREENING*
- *AUTOMATED ACQUISITION*
- *AUTOMATED ANALYSIS*
- *HIGH RESOLUTION*



The ImageXpress™ *5000A automated cellular imaging and analysis system is an integrated hardware and software system designed specifically for rapid acquisition and analysis of fluorescence images for high throughput cell-based screening.*

Why choose *ImageXpress*?

Instrumentation

ImageXpress is ***fast.*** Scanning an entire 384-well plate at full resolution (1280 × 1024 pixels), with two fluorescence images per well, takes about 20 minutes. *ImageXpress* achieves this level of performance because the entire system—from the custom-designed optical components to the efficient software user interface—is **designed and built for screening.** Axon's optical, mechanical and electronics engineers have created a fully-automated inverted epifluorescence imaging system with the features you need:

- **efficient,** uniform illumination for brief exposure times and maximal throughput
- **robust,** high-speed laser autofocus
- **rapid** selection of up to 10 excitation and emission filters and up to 4 dichroic beamsplitters
- **reliable,** high-speed, precision mechanics for continuous operation
- **uncompromised** image quality, with diffraction-limited imaging optics and high-resolution cooled CCD imaging.

Software

ImageXpress includes a **complete software solution for screening—at no extra cost.** All phases of your workflow are included, from interactive control of the instrument during assay development, to automated plate scanning, data storage and analysis. Powerful visualization tools are included that link the extracted data in spreadsheets and custom graphs to the actual objects in images for quality control. Our software engineers have built a system with features every bit as impressive as the hardware:

- **powerful,** intuitive graphical user interface
- **scalable,** high-performance, client-server database for image storage, retrieval and archiving
- **full scripting** capability for maximally flexible automation
- **sophisticated** image analysis algorithms for automated analysis
- **refreshingly,** the *ImageXpress* software suite is included in the base system price.

The *ImageXpress* system has the power and speed to run your current assays efficiently. With its open-system architecture and full scripting capability, *ImageXpress* software provides unmatched flexibility to speed your assay development team in creating your next generation of high-information content fluorescent cellular assays.

Support

Axon Instruments stands behind *ImageXpress* with responsive, free, **telephone and e-mail technical support**, two days of **training** at the time of installation, **software updates, on-site preventive maintenance and field service** for the first year. An optional extended service agreement is also available.

Finally, the *ImageXpress* system is reasonably priced. Read on for further details about the system, and contact us to arrange a demonstration.

A Cellular Imaging System Designed for Automated, High-Resolution Screening

The *ImageXpress* optical system was designed to increase throughput for screening without compromising image quality. It uses a full-spectrum high-power xenon arc lamp, custom light guide, and Abbé illumination optics to deliver efficient, uniform illumination to the sample, and keep exposure times to a minimum.

Reliable components for continuous operation

The system uses high-quality, infinity-corrected Nikon objective lenses, mounted in a motorized turret (holds up to 6 objectives). The motorized excitation and emission filter wheels (10 positions each) and motorized dichroic beamsplitter wheel (4 positions) accept standard filter sets for maximum

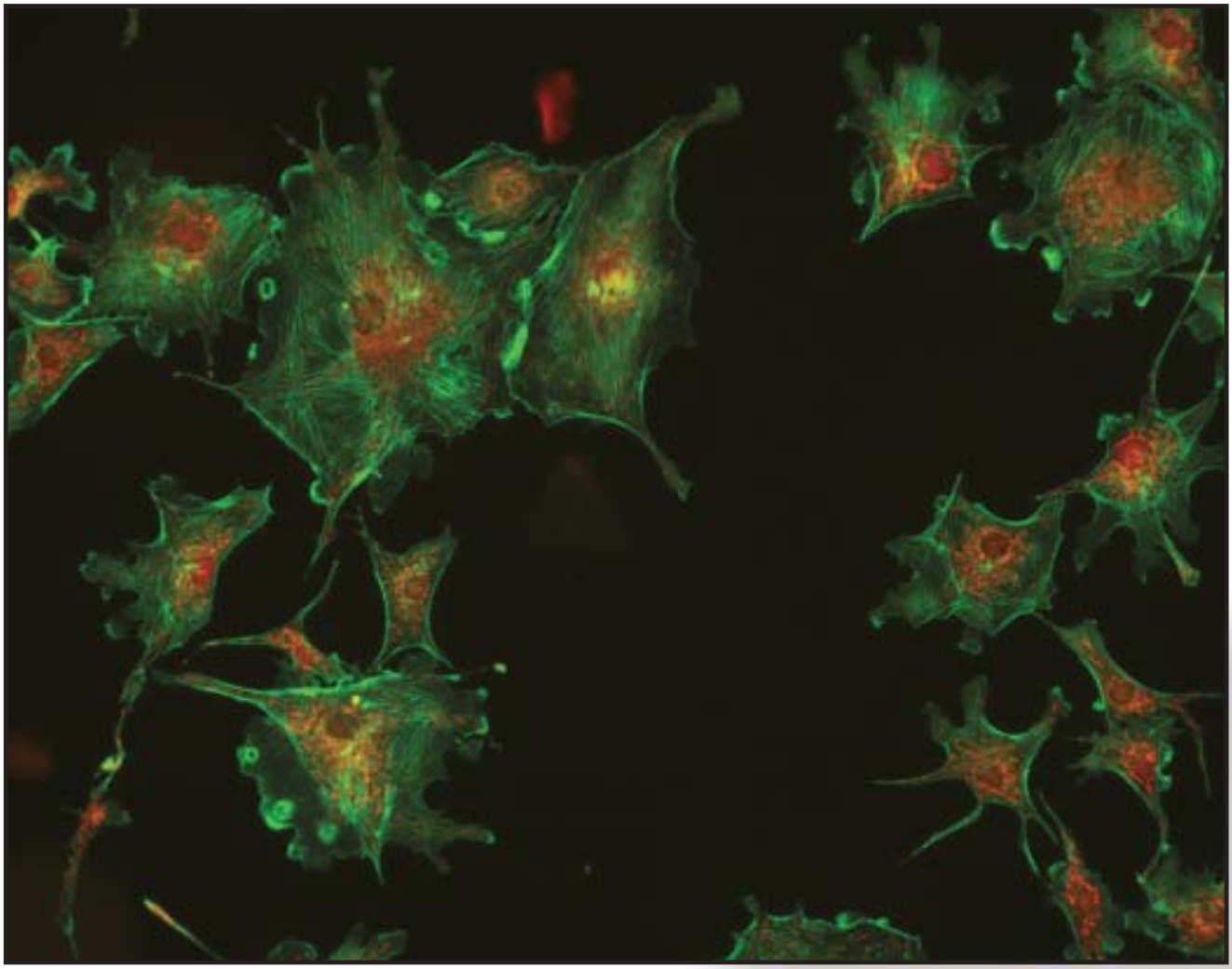


Figure 1 Pseudocolor composite image of triple-labeled cells acquired with ImageXpress. *Shown are bovine pulmonary artery epithelial cells stained with DAPI (cell nuclei, blue channel), BODIPY®-FL phallacidin (F-actin, green channel) and MitoTracker® Red CMXRos (mitochondria, red channel). From a FluoCells® prepared slide, Molecular Probes, Eugene, OR.*

flexibility. The precision motorized stages (X-Y stage for plate motion and Z stage for focusing) are designed for rapid movement and trouble-free, continuous operation. Encoders on the X-Y and Z stages ensure accurate, consistent positioning. All motion control operates asynchronously, ensuring fast setup for each image acquisition and therefore increased throughput.

High-resolution imaging of all standard plate formats

Images of the cells (Figure 1) are digitized by a megapixel cooled CCD camera designed and manufactured by Axon Instruments specifically for this instrument. The system is designed for scanning multiwell microplates (96-, 384-, or 1536-well formats), but can accommodate any sample that will fit within the footprint of a microplate (e.g., standard microscope slides or petri dishes, in an appropriate holder).

High-speed laser autofocus

Because microplates are not optically flat, the ImageXpress microscope uses a high-speed laser autofocus mechanism to focus on the bottom of each well. The user can additionally specify offsets from this surface (e.g., for focal shift of the objective with emission wavelength or for cells grown on top of a feeder layer). The laser-based autofocus has several advantages over image-based autofocus:

- laser-based autofocus is much faster
- virtually no significant bleaching of the sample

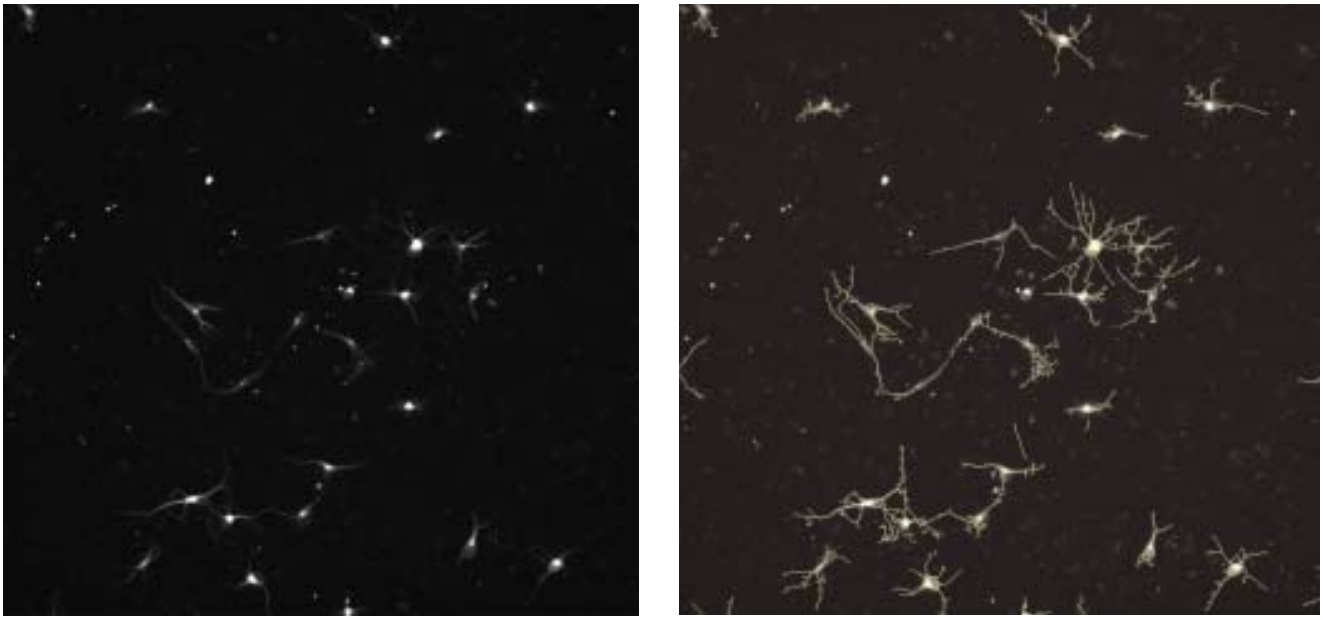


Figure 7 ImageXpress *features a sophisticated cellular image analysis toolbox including neurite detection and measurement. On the left is an image of primary neural cultures stained for tubulin. On the right is the same image with an overlay of the skeletonized neurites automatically detected by* ImageXpress.

Sophisticated cellular image analysis toolbox

Automated image analysis tools include:

- Flat-field and background correction algorithms
- Real-time cell counting algorithm
- Cell nucleus detection and measurement
- Cytoplasmic fluorescence measurements
- Neurite detection and measurement
  - linear feature detection
  - length measurement
  - neurite tree decomposition (identification of individual neurite segments, branch points)
  - association of neurites with cell bodies
  - complexity measurement.

Available image feature measurements include:

- object pixel statistics: average intensity, median, variance, etc.
- area
- perimeter
- best-fit ellipse
- angle of major axis
- bounding box
- center of mass
- convex hull
- texture characterization.

The scripting interface gives you the flexibility to create any number of derived measurements, for example:

- form factor
- aspect ratio
- convexity
- equivalent circle diameter.

Open architecture and full scripting capability mean extensibility

ImageXpress software is an open system: we provide complete documentation of the scripting interface (including the image analysis functions) so that you can create your own specialized scripts, whether for acquisition, analysis or a combination of both. In addition, we have an ongoing collaboration with the internationally-recognized CSIRO (Commonwealth Science and Industry Research Organization, Australia) Mathematical and Information Sciences image analysis group to continue extending the image analysis functionality of the program. Finally, assistance is available for developing custom scripts for your assays, on a contract basis.

Powerful visualization tools connect analysis results to images

Numerical image analysis results are presented in spreadsheet format, with each row representing an object in an image, and each column representing a measured feature or property of that object. The *ImageXpress* console (Figure 8) gives you three linked views of the data:

1. a spreadsheet of analysis results
2. graph windows for creating scatterplots and histograms
3. the actual objects in the images.

The combination of image, spreadsheet and graphical views is a powerful tool for visualizing, classifying, sorting, saving and exporting your results.

With the results spreadsheet you can:

- Sort the data on any column or combination of columns
- Select columns for scatterplot or histogram graphs
- Select or exclude results based on threshold criteria that you establish
- Export your results to any spreadsheet or database program

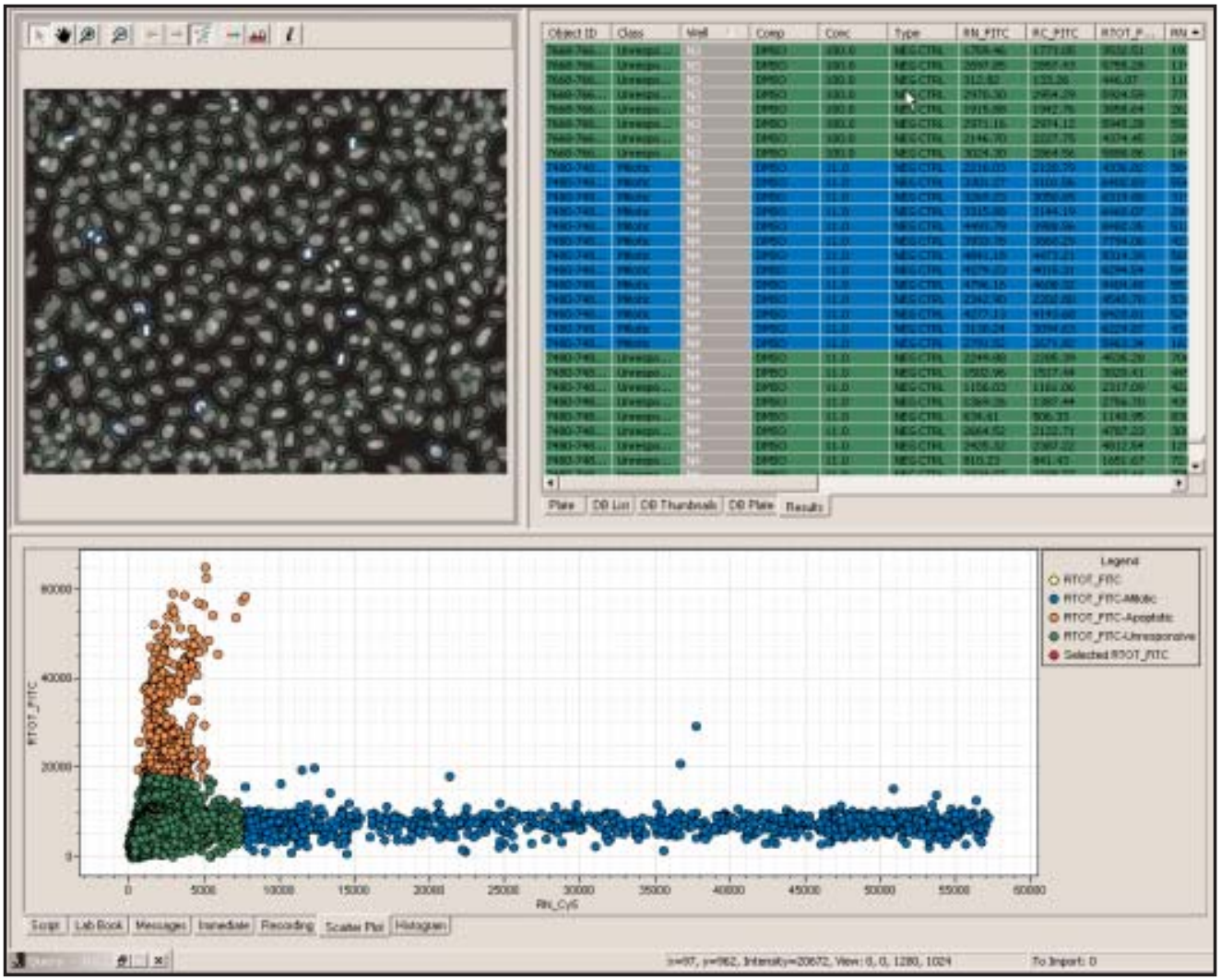


Figure 8 Powerful linked results views connect the analyzed image features with the results spreadsheet and data points in graphs. The graph shows results from an apoptosis vs. mitosis assay (see Figure 9 for a description of the staining). Each point in the scatterplot represents a cell. The measured annexin response RTOT_FITC) is plotted against the measured mitotic indicator response (RN_Cy5). The ImageXpress *automated classification algorithm has been used to color-code cells by response: apoptotic cells are orange, mitotic cells blue, and unresponsive cells are green. Note that the color coding is reflected in all three data views.*

Features

An optomechanical system designed to increase throughput for screening

- High-efficiency, custom imaging optics
- High-speed laser autofocus
- Image-based autofocus for unusual imaging situations
- Powerful xenon light source
- Efficient, uniform illumination to minimize exposure times
- Fast, precise X-Y stage
- Asynchronous control of all components increases throughput
- High-resolution (1280 × 1024 pixels) cooled CCD imaging
- Available binning modes decrease exposure time and increase throughput
- Uncompromised image quality
- Optional robotic plate handling.

Software that is powerful, flexible and scalable, but still easy-to-use

- Intuitive graphical user interface
- One-click interactive instrument controls for assay development
- Modular client-server software architecture provides maximal flexibility, scalability
- Fully scriptable: any software function can be scripted for walk-away automation
- Acquisition and Analysis Script Wizards automate script writing
- Images automatically annotated with complete acquisition information
- Any number of user-specified annotations can be added
- Annotated images automatically stored to database
- Network-enabled database allows images to be stored on a central server
- Retrieve images with configurable tree query or database query dialog
- Export images to standard TIFF files for publication or use in other programs
- Complete biological image analysis toolbox
- Automated classification of objects
- Analysis algorithms developed in collaboration with CSIRO Image Analysis Group.

Service and Support

- Installation and two-days of on-site training included
- Additional on-site or web-based training available (call for details)
- Custom script programming services available (call for details)
- The same quality, PhD-level technical support you've come to expect from Axon Instruments
- One year warranty and on-site service contract included
- Optional extended on-site service contract available.

- ***Fluorescence assays probe a huge range of cellular functions, including:***
  - apoptosis
  - cell viability
  - cell classification and counting
  - neurite outgrowth
  - cytoskeletal rearrangements
  - subcellular localization
  - molecular co-localization
  - intracellular trafficking
  - receptor translocation
- ***Cell-based assays can identify toxic compounds earlier in the discovery cycle.***
- ***High-resolution imaging enables investigations of heterogeneity of response among cell types.***

With the *ImageXpress* system, these cellular functions can be measured and analyzed with maximum efficiency. The *ImageXpress* system was designed expressly for screening, and combines optimized custom optics, precision mechanics, powerful but easy-to-use control software, sophisticated image assays provided at no extra charge, and a practically limitless database for storage and retrieval of images. The *ImageXpress* system's open software interface and full scripting capability deliver unmatched flexibility for creating customized assays.

C.2 Polartechnics Solarscan

The second document is an independent and objective assessment of the state of the Polartechnics Solarscan industrial project, on which I worked between 1994 and 2004, by the Australian government. This assessment was conducted in 2005, shortly after I left Australia.

Australian Government

Department of Health and Ageing

Australia and New Zealand Horizon Scanning Network

ANZHSN

AN INITIATIVE OF THE NATIONAL, STATE AND TERRITORY GOVERNMENTS OF AUSTRALIA AND THE GOVERNMENT OF NEW ZEALAND

National Horizon Scanning Unit
Horizon scanning prioritising summary

Volume 8, Number 1:

Solar Scan for the detection and monitoring of melanoma.

February 2005

AHTA

Adelaide Health Technology Assessment

Electronic copies can be obtained from http://www.horizonscanning.gov.au

Enquiries about the content of this summary should be directed to:

HealthPACT Secretariat
Department of Health and Ageing
MDP 106
GPO Box 9848
Canberra ACT 2606
AUSTRALIA

The production of this *Horizon scanning prioritising summary* was overseen by the Health Policy Advisory Committee on Technology (HealthPACT), a sub-committee of the Medical Services Advisory Committee (MSAC). HealthPACT comprises representatives from health departments in all states and territories, the Australia and New Zealand governments; MSAC and ASERNIP-S. The Australian Health Ministers' Advisory Council (AHMAC) supports HealthPACT through funding.

This *Horizon scanning prioritising summary* was prepared by Adriana Parrella from the National Horizon Scanning Unit, Adelaide Health Technology Assessment, Department of Public Health, Mail Drop 511, University of Adelaide, South Australia, 5005.

PRIORITISING SUMMARY

REGISTER ID: **000136**

NAME OF TECHNOLOGY: **SOLARSCAN®**

PURPOSE AND TARGET GROUP: **MELANOMA DETECTION AND MONITORING SYSTEM FOR ROUTINE SKIN CHECKS**

STAGE OF DEVELOPMENT (IN AUSTRALIA):

☐	Yet to emerge	☒	Established
☐	Experimental	☐	Established *but* changed indication or modification of technique
☐	Investigational	☐	Should be taken out of use
☐	Nearly established		

AUSTRALIAN THERAPEUTIC GOODS ADMINISTRATION APPROVAL

☐	Yes	ARTG number	
☒	No	☐	Not applicable

INTERNATIONAL UTILISATION:

COUNTRY	LEVEL OF USE		
	Trials Underway or Completed	Limited Use	Widely Diffused
Australia		✓	

IMPACT SUMMARY:

Polartechnics Ltd provides SolarScan® with the aim of detecting melanoma and monitoring skin lesions. The technology is currently available through several general practice or dermatology clinics for people requiring skin lesion monitoring and/or detection of melanoma within Australia.

BACKGROUND

Dermoscopy (surface microscopy) is the clinical technique used to examine skin lesions. It involves using a hand-held magnifying instrument (10 x magnification), usually with liquid at the skin-instrument interface, to examine pigmented lesions arising on the skin surface. This technique allows the observer to look not only onto but also into the superficial skin layers, and thus permits a more detailed inspection of pigmented skin lesions (Crotty and Menzies 2004; Kittler et al 2002). Dermoscopy assists the clinician to determine whether a skin lesion requires excision, biopsy, monitoring or can be safely left *in situ*. It is possible that it increases the accuracy of melanoma detection when compared to standard visual inspection (Crotty and Menzies 2004).

The SolarScan® device was developed by Polartechnics Ltd., CSIRO and the Sydney Melanoma Unit. It consists of a remote head colour video camera that produces high resolution (24-bit, 760 x 570-pixel) images. The lesion image is digitised for processing. The device uses surface epiluminescence microscopy, which allows for x40 magnification (Figure 1) (Polartechnics 2004).

The SolarScan® takes digitised images of lesions and extracts the lesion characteristics, which are then compared to a database of benign and malignant lesions.

The SolarScan® can detect melanomas less than 3mm deep which may allow for early detection and treatment. The technology is also designed to monitor any changes in lesions over time.

Figure 1. Solarscan (Printed with permission Polartechnics)

CLINICAL NEED AND BURDEN OF DISEASE

In Australia there were 8,885 new cases of skin melanoma recorded in the year 2001, a rate of 45.8 per 100, 000 (AIHW 2004a). In 2001, melanoma was the fourth most common cancer in Australia and accounted for 10% of all new cancer cases (AIHW 2004).

Incidence data for cancers of the skin, apart from melanoma, are not collected on a routine basis by cancer registries. These common cancers are not legally notifiable and are therefore not routinely reported. Estimates of the frequency of treated skin cancers, ie basal cell carcinoma and squamous cell carcinoma, are derived from data that have been collected in national household surveys in 1985, 1990, 1995 and 2002 (NCCI 2003). A 2002 national survey found 374,000 people had been diagnosed with either squamous or basal cell carcinoma in Australia compared to 270,000 in 1995 (Cancer Council Victoria 2004).

DIFFUSION

There are currently 40 SolarScan® machines installed in general practice and/or dermatology clinics in Australia (personal communication Polartechnics).

COMPARATORS

The comparators for skin lesion inspection and monitoring are visual inspection by the skin clinician or general practitioner using, as mentioned previously, a handheld surface microscope (dermoscope).

EFFECTIVENESS AND SAFETY ISSUES

See complete volume of Prioritising Summaries for definitions of Levels of Evidence.
At the time of preparing this summary, the manufacturer was in the process of submitting a paper for publication describing a trial of the SolarScan® and its diagnostic accuracy compared to dermoscopy experts and general practitioners (personal communication, Polartechnics).

The study by Menzies et al (2001) (level III-3 diagnostic evidence) demonstrated the effectiveness of SolarScan® in early detection of clinically "featureless" melanoma. The measurement outcome was the specificity of melanoma diagnosis for short-term digital surface microscopic monitoring of suspicious or changing atypical melanocytic lesions. 318 consecutive lesions from 245 patients (aged 4 – 81 years) were monitored during a 2.5 to 4.5 month period.

Of the 318 lesions, 257 (81%) remained unchanged and 61 (19%) showed morphologic changes. Of the 61 lesions that changed, 7 were found to be early melanoma (11% of all changed lesions, 2% of total lesions): 2 invasive lesions and 5 *in situ*. The authors report that none of the melanomas developed any classic surface microscopic features of melanoma on examination with a handheld surface microscope and could be identified only by morphologic change. The specificity of the SolarScan® was 83% when compared to pathology results of the excised lesions.

There are no studies, as yet, that assess the impact of possible early melanoma diagnosis with the SolarScan®, compared to visual inspection or dermoscopy on the health outcomes (ie survival) of patients.

COST IMPACT
There are several MBS item numbers for the removal of basal and squamous cell carcinoma (item numbers 31255 – 31295) with fees ranging from $190.00 - $240.00 each and for the removal of malignant melanoma (item numbers 31300 – 31335) at a cost ranging from $224.00 - $315.00 (MBS 2004). The current cost of the SolarScan® device is approximately $30,000.

The total number of public hospital separations in Australia for malignant melanoma or other malignant neoplasm of the skin was 82,707 during the year 2002-03 (AR-DRG numbers C43 and C44). In addition the number of public hospital separations for melanocytic naevi and benign neoplasms of the skin (AR-DRG numbers D22 and D23) were 10,837 and 5,332 for the same time period.

There are currently high rates of skin lesion excisions; in particular, there are high numbers of benign lesions excised compared to malignant lesions. The high excision rates occur because it is common for pigmented skin lesions such as naevi and seborrhoeic keratoses to appear similar to melanoma. It has been shown that there are approximately 11-29 benign excisions per malignant excision and up to 36 excisions per malignant excision when seborrhoeic keratoses are included (English et al 2004).

If the SolarScan® device demonstrates more accurate diagnosis than visual inspection or hand held surface microscope (currently not available), it may potentially reduce the number of unnecessary surgical procedures for the excision of suspect melanomas and therefore pathology costs.

ETHICAL, CULTURAL OR RELIGIOUS CONSIDERATIONS
No issues were identified/raised in the sources examined.

OTHER ISSUES
No issues were identified/raised in the sources examined.

CONCLUSION:
There is the potential for this technology to benefit a large number of patients, based on the high burden of skin cancer and the cost of detecting and treating melanoma and other skin cancers in the Australian population. However, the safety and effectiveness of this technology cannot be determined until further studies with the SolarScan® are published.

HEALTH PACT ACTION:
Technology is already diffusing into the Australian health system and will not impact significantly in terms of policy or cost burden. Archive.

SOURCES OF FURTHER INFORMATION:

AIHW (2004a). *'Interactive cancer data'* [Internet]. Australian Institute of Health and Welfare. Available from: http://www.aihw.gov.au/cognos/cgi-bin/ppdscgi.exe?DC=Q&E=/Cancer/canceragerates v7 [Accessed 24th November, 2004].

AIHW (2004b). *'Cancer in Australia 2001'* [Internet]. Australian Institute of Health and Welfare. Available from: http://www.aihw.gov.au/publications/can/ca01/ca01.pdf [Accessed 24th November, 2004].

Cancer Council Victoria (2004). *'Skin cancer continues to increase burden on the health system'* [Internet] Available from: [Accessed 24th November, 2004].

Aitken, J. F., Janda, M. et al (2004). 'Prevalence of whole-body skin self-examination in a population at high risk for skin cancer (Australia)', *Cancer Causes Control,* 15 (5), 453-463.

Argenziano, G., Soyer, H. P. et al (2003). 'Dermoscopy of pigmented skin lesions: results of a consensus meeting via the Internet', *J Am Acad Dermatol,* 48 (5), 679-693.

Crotty, K. A. & Menzies, S. W. (2004). 'Dermoscopy and its role in diagnosing melanocytic lesions: a guide for pathologists', *Pathology,* 36 (5), 470-477.

English, D. R., Del Mar, C. & Burton, R. C. (2004). 'Factors influencing the number needed to excise: excision rates of pigmented lesions by general practitioners', *Med J Aust,* 180 (1), 16-19.

Gutenev, A., Skladnev, V. N. & Varvel, D. (2001). 'Acquisition-time image quality control in digital dermatoscopy of skin lesions', *Comput Med Imaging Graph,* 25 (6), 495-499.

Janda, M., Elwood, M. et al (2004a). 'Prevalence of skin screening by general practitioners in regional Queensland', *Med J Aust,* 180 (1), 10-15.

Janda, M., Youl, P. H. et al (2004b). 'Attitudes and intentions in relation to skin checks for early signs of skin cancer', *Prev Med,* 39 (1), 11-18.

Kittler, H., Pehamberger, H. et al (2002). 'Diagnostic accuracy of dermoscopy', *Lancet Oncol,* 3 (3), 159-165.

Menzies, S. W., Gutenev, A. et al (2001). 'Short-term digital surface microscopic monitoring of atypical or changing melanocytic lesions', *Arch Dermatol,* 137 (12), 1583-1589.

National Cancer Control Initiative (2003). *The 2002 national non-melanoma skin cancer survey.* A report by the NCCI Non-melanoma Skin Cancer Working Group. Melbourne: National Cancer Control Initiative.

Polartechnics (2004). *'SolarScan'* [Internet] Available from: http://www.polartechnics.com/Products/SolarScan/SolarScan.htm [Accessed 24th November, 2004].

Rosado, B., Menzies, S. et al (2003). 'Accuracy of computer diagnosis of melanoma: a quantitative meta-analysis', *Arch Dermatol,* 139 (3), 361-367; discussion 366.

SEARCH CRITERIA TO BE USED:

Carcinoma, Basal Cell/diagnosis/pathology
Melanoma/classification/ diagnosis/ pathology
Microscopy/ methods/standards
Self-Examination/ utilization
Skin Neoplasms/ diagnosis/etiology

Appendix D

Directional Morphological Filtering

This article was published in IEEE TPAMI as Soille and Talbot (2001).

```
@article{SoilleTalbot2001,
Author = {Soille, P. and Talbot, H.},
Journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
Number = {11},
Pages = {1313-1329},
Title = {Directional Morphological Filtering},
Volume = {23},
Year = {2001}}
```

Directional Morphological Filtering

Pierre Soille and Hugues Talbot

Abstract—We show that a translation invariant implementation of min/max filters along a line segment of slope in the form of an irreducible fraction dy/dx can be achieved at the cost of $2+k$ min/max comparisons per image pixel, where $k=\max(|dx|,|dy|)$. Therefore, for a given slope, the computation time is constant and independent of the length of the line segment. We then present the notion of periodic moving histogram algorithm. This allows for a similar performance to be achieved in the more general case of rank filters and rank-based morphological filters. Applications to the filtering of thin nets and computation of both granulometries and orientation fields are detailed. Finally, two extensions are developed. The first deals with the decomposition of discrete disks and arbitrarily oriented discrete rectangles, while the second concerns min/max filters along gray tone periodic line segments.

Index Terms—Image analysis, mathematical morphology, rank filters, directional filters, periodic line, discrete geometry, granulometry, orientation field, radial decomposition.

1 Introduction

Similar to the perception of the orientation of line segments by the human brain [1], [2], computer image processing of oriented image structures often requires a bank of directional filters or template masks, each being sensitive to a specific range of orientations [3], [4], [5], [6], [7], [8]. In this context, recent advances in the fast implementation of morphological filters along discrete lines at arbitrary angles [9] have opened avenues for new image analysis applications [10], [11], [12], [13]. However, if one needs to obtain results invariant to translations of the image frame, the algorithm proposed in [9] is not suitable because the shape of the structuring element (SE) varies slightly from one pixel position to another. This effect is not due to image border conditions but to the fact that the structuring element is defined by considering n consecutive pixels along the considered discrete connected line. Therefore, the shape depends on the position of the structuring element along the line (except for lines matching one of the principal directions of the image grid).

In this paper, we address this problem and demonstrate that a fast translation invariant (TI) implementation can be achieved at the cost of a few extra computations and still independently of the length of the SE. Therefore, it is much more efficient than the raw TI algorithm which has a linear computational complexity. In addition, we show that a similar performance can be obtained for the more general class of rank filters.

The paper is organized as follows: Section 2 describes the problem and illustrates it on a real example. Strategies allowing for a translation invariant implementation for both min/max and rank directional filters are proposed in Section 3. Performance evaluation and comparison with other approaches are carried out in Section 4. Applications to the filtering of thin networks and the computation of linear granulometries and orientation fields are presented in Section 5. Before concluding, extensions to line segment cascades and gray-tone line segments of constant slope are presented in Section 6.

2 Problem Description

An efficient algorithm for computing erosions and dilations along discrete Bresenham lines [14] at arbitrary orientations has been proposed in [15], [9]. The processing of a given line is based on the recursive procedure detailed in [16], [17] and requiring only three min/max comparisons per pixel, whatever the length of the SE (see further improvements in the case of independent, identically distributed 1D signals [18]). However, as already noticed in [9], for orientations which do not correspond to one of the principal directions of the digitisation grid, the shape of the line segment varies slightly from one pixel to another. For instance, Fig. 1 shows that, when computing min/max filters along a discrete Bresenham line of slope $-1/3$, the shape of the SE varies along the line, and this variation is periodic with a periodicity equal to three. It follows that the output of filters computed along a discrete Bresenham line are not translation invariant.

Since we are dealing with images digitized on a square grid, we can restrict our analysis to line slopes in the form of an irreducible fraction dy/dx (i.e., dx and dy are integers with no common divisors). By convention, it is convenient to include the forms $0/1$ and $1/0$ for referring to horizontal and vertical lines, respectively. Now, for a Bresenham line of slope in the form of an irreducible fraction dy/dx, there are $\max(|dx|,|dy|)$ different line segments defined along the discrete line. This number corresponds to the periodicity $k=\max(|dx|,|dy|)$ of the elementary pattern occurring along the Bresenham line. In the sequel, we denote by $L_{\lambda_i,(dx,dy)}$ the connected line segment obtained by considering λ successive pixels of a Bresenham line of slope dy/dx, starting from the ith pixel of the line ($i \in \{1,\ldots,k\}$). For

- *P. Soille is with the EC Joint Research Centre, Space Applications Institute, T.P. 262, I-21020 Ispra, Italy. E-mail: Pierre.Soille@jrc.it.*
- *H. Talbot is with CSIRO-Mathematical and Information Sciences, Image Analysis Group, Locked Bag 17, North Ryde (Sydney), NSW 2113, Australia. E-mail: Hugues.Talbot@cmis.csiro.au.*

Manuscript received 9 Dec. 1999; revised 17 Nov. 2000; accepted 28 Feb. 2001.
Recommended for acceptance by A. Kundu.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 111062.

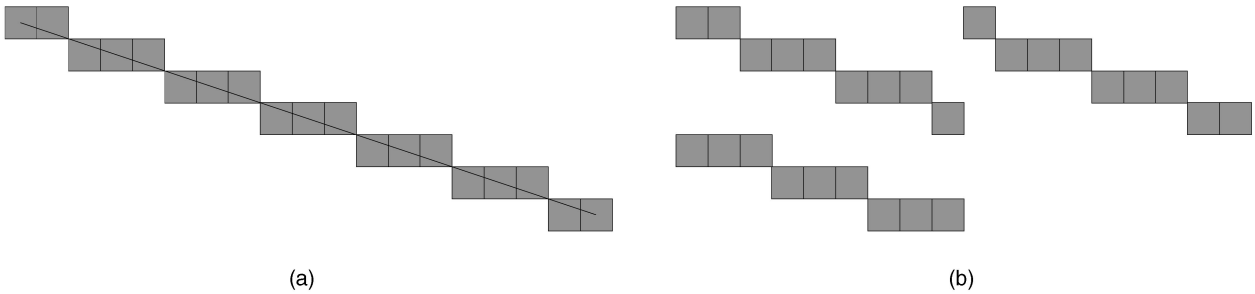


Fig. 1. Line segment patterns along a discrete Bresenham line. (a) A discrete Bresenham line of slope $-1/3$. The line has been drawn from the top left pixel (i.e., "origin" of the line). (b) The three line segment SEs occurring along the discrete line (here, for a length of nine pixels).

example, Fig. 1b shows the three line segments of length equal to nine pixels for a slope of $-1/3$: $L_{9_1,(3,-1)}$ (top left SE), $L_{9_2,(3,-1)}$ (top right SE), and $L_{9_3,(3,-1)}$ (bottom SE).

A real example illustrating the non-TI property of the algorithm described in [9] is given in Fig. 2. We define a two-dimensional (2D) gray-scale image f as a function which maps a discrete grid of 2D pixel coordinates to a finite chain of integers. The input gray-scale image (Fig. 2a) shows a plant leaf. The detection of the leaf veins for a given range of orientations can be achieved with a series of directional openings. This is a preprocessing step for the automatic detection and separation [19] of the plants as seen from an autonomous vehicle navigating in a field [20], [21], [22]. For example, the opening γ by a line segment of given slope and length will remove all veins except those having this orientation and at least this length. However, because the veins consist of a network of thin, mostly one pixel-thick, bright objects, the output of the non-TI implementation of the opening varies significantly when shifting the image frame, as illustrated in Figs. 2b, 2c, 2d, 2e, and 2f. In this example, there are five possible outputs because the periodicity of a Bresenham line of slope $-1/5$ equals five. For those pixel positions where the structuring element was only partially fitting the image frame, we have assumed that pixels outside the image definition domain are all set to zero (this is highlighted in Fig. 3 where all intermediate steps leading to an opening are displayed).

In general, there are k possible outputs for a neighborhood image operator Ψ by a line segment of length λ pixels applied along a Bresenham line of slope dy/dx and sweeping the

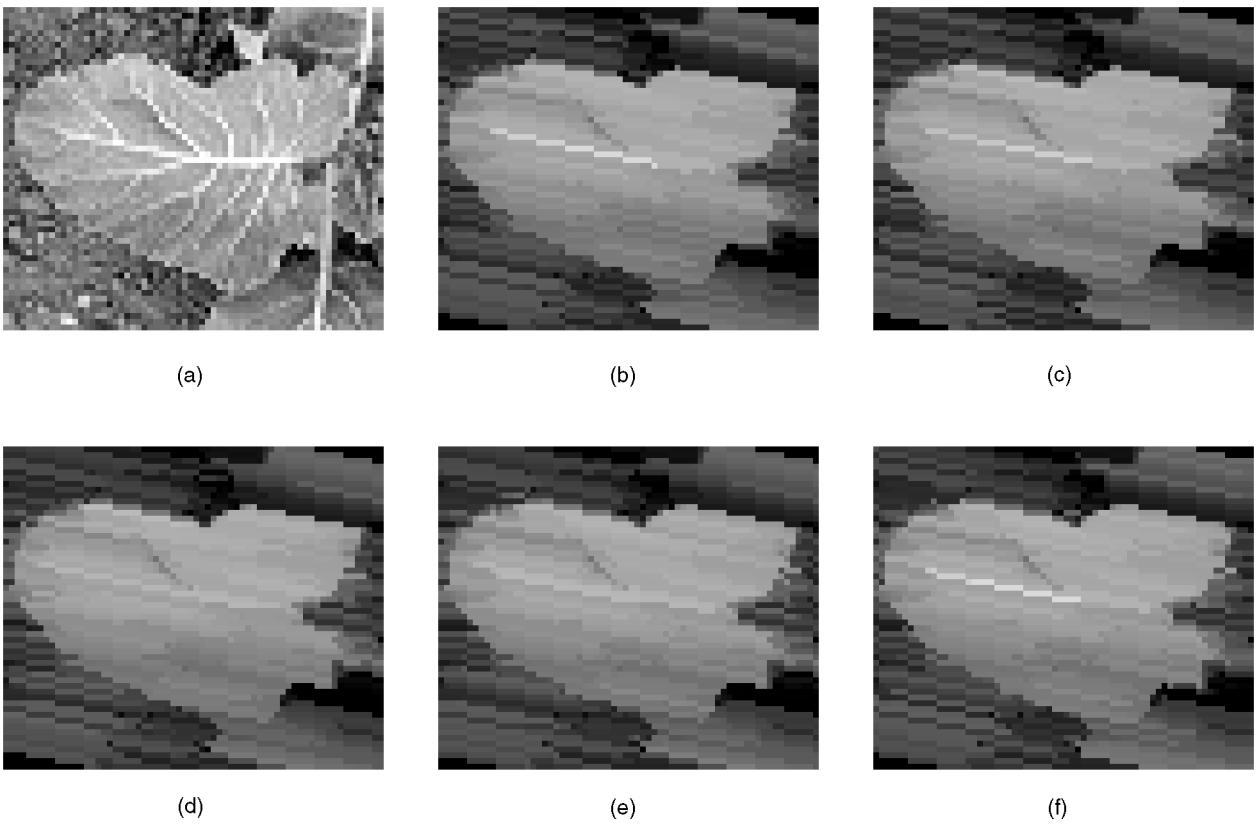


Fig. 2. Opening γ by a line segment of 15 pixels along Bresenham lines of slope $-1/5$. (a) Input image f: A 66×56 image of a cauliflower leaf (cropped region of an outdoor scene [19]). (b) $\gamma_{BL_{15_1},(5,-1)}(f)$, (c) $\gamma_{BL_{15_2},(5,-1)}(f)$, (d) $\gamma_{BL_{15_3},(5,-1)}(f)$, (e) $\gamma_{BL_{15_4},(5,-1)}(f)$, and (f) $\gamma_{BL_{15_5},(5,-1)}(f)$ are the five possible outputs using the recursive algorithm described in [9].

whole image definition domain. We denote them by $\Psi_{BL_{\lambda_i},(dx,dy)}$, where $i \in \{1,\ldots,k\}$ and $k = \max(|dx|,|dy|)$.

3 Proposed Translation Invariant (TI) Algorithms

After introducing few background notions, we show that the recursive non-TI algorithm for min/max filters [9] as well as the so-called moving histogram algorithm [23], [24] for rank filters can be adapted so as to lead to efficient TI implementations.

3.1 Background Notions

In [25], we have shown that the convex hull transformation can be defined in terms of an intersection (point-wise minimum $\wedge$) of half-plane closings. More precisely, denoting by ϕ the closing transformation, π_θ a closed half-plane having a given slope $\theta = \arctan(dy/dx)$, and $\check{\pi}_\theta$ the reflected half-plane, the convex hull transformation CH of a gray-scale image f is defined as follows:

$$CH(f) = \bigwedge_\theta [\phi_{\pi_\theta}(f) \wedge \phi_{\check{\pi}_\theta}(f)]. \quad (1)$$

As demonstrated in [26], the translation-invariant implementation of half-plane closings[1] can be achieved by processing the pixels in the order they are reached when progressively translating each Euclidean half-plane π_θ so as to sweep the whole image definition domain. By doing so, the pixels reached at any given step correspond to those falling along the periodic line [28] having the same slope as the half-plane. However, since we aim at computing min/max filters along a *connected* discrete line segment, it is not enough to apply them along periodic lines. Two different strategies for circumventing this problem are investigated in Sections 3.2 and 3.3 (the first for min/max and the second for rank filters).

3.2 Recursive TI Algorithms

The adaptation of the recursive non-TI algorithm so as to lead to recursive TI min/max filters is developed in Section 3.2.1. Section 3.2.2 shows that, if we are interested in computing directional openings and closings, appropriate combinations of a series of non-TI openings or closings directly lead to TI openings and closings.

3.2.1 Erosions and Dilations

We define the slope of the SE in the form of an irreducible fraction dy/dx. We assume that the length λ (in pixels) of the connected SE equals n times the periodicity k of the corresponding discrete Bresenham line, i.e., $\lambda = nk$ and $k = \max(|dx|,|dy|)$. The underlying periodic line segment is denoted by $P_{n,\vec{v}}$ and defined as follows:

$$P_{n,\vec{v}} = \bigcup_{i=0}^{n-1} i\vec{v}, \quad (2)$$

where $n \geq 1$ is the number of points in the periodic line segment and $\vec{v} = (dx,dy)$ is a constant vector separating all successive points along the periodic line segment.[2] We then apply the following two step algorithm:

1. Compute the min/max filters with the periodic line segment $P_{n,(dx,dy)}$.
2. Select an arbitrary line segment $L_{k_i,(dx,dy)}$ of length k among all k possible occurring along the Bresenham discrete line of slope dy/dx and uses it to compute the min/max filters of the image produced by the first step. Note that, although it is convenient to consider one of the k possible eight-connected discrete patterns occurring along the Bresenham discrete line, any connected line joining $(0,0)$ to (dx,dy), but excluding this latter pixel, could be considered instead.

Hence, the recursive TI strategy relies upon the following SE decomposition:

$$L_{nk_i,(dx,dy)} = L_{k_i,(dx,dy)} \oplus P_{n,(dx,dy),} \quad (3)$$

where $\oplus$ denotes Minkowski addition [29] and $i \in \{1,\ldots,k\}$. Denoting dilation by δ and erosion by ε, the following relationships hold:

$$\delta_{L_{nk_i,(dx,dy)}} = \delta_{L_{k_i,(dx,dy)}} \delta_{P_{n,(dx,dy),}} \quad (4)$$

$$\varepsilon_{L_{nk_i,(dx,dy)}} = \varepsilon_{L_{k_i,(dx,dy)}} \varepsilon_{P_{n,(dx,dy)}}. \quad (5)$$

Figs. 3b, 3c, 3d, and 3e detail the successive steps for computing the recursive TI opening by $L_{15_4,(5,-1)}$ of the image shown in Fig. 3a:

$$\underbrace{\gamma_{L_{15_4,(5,-1)}}(\overbrace{f}^{\text{Fig. 3a}})}_{\text{Fig. 3e}} = \underbrace{\delta_{\check{L}_{5_4,(5,-1)}}(\delta_{\check{P}_{3,(5,-1)}}(\underbrace{\varepsilon_{L_{5_4,(5,-1)}}(\overbrace{\varepsilon_{P_{3,(5,-1)}}(f)}^{\text{Fig. 3b}})}_{\text{Fig. 3c}}))}_{\text{Fig. 3d}}.$$

As emphasized in [9], the min/max filter with a periodic SE $P_{n,(dx,dy)}$ requires three min/max comparisons per pixel whatever the values of n and the slope dy/dx. The brute force (raw) approach for computing the min/max filter with $L_{k_i,(dx,dy)}$ requires $k-1$ min/max comparisons per pixel. Less than k additional computations can be obtained in situations where a scanning order of the image pixels producing a high degree of overlap between two successive positions of $L_{k_i,(dx,dy)}$ can be found. Therefore, the maximum total number of $2+k$ min/max operations per pixel for a TI implementation is independent of the length of the SE and varies according to the periodicity k of the underlying periodic line segment.

3.2.2 Openings and Closings

Alternatively, when computing openings (or closings), the non-TI recursive implementation of min/max filters can be combined so as to lead to a TI implementation. Indeed, because the plain line segments $L_{\lambda_i,(dx,dy)}$ are defined along

1. The ANSI C code for the non-TI implementation is given in [27].

2. Note that periodic lines were originally defined [28] as follows: $P_{\lambda,\vec{v}} = \cup_{i=0}^{i=\lambda} i\vec{v}$. Here, we have adapted the definition of periodic lines so that λ equals the number of pixels of the periodic line (rather than $\lambda+1$ in the original definition) in accordance with the parameter λ used for Bresenham connected line segments L.

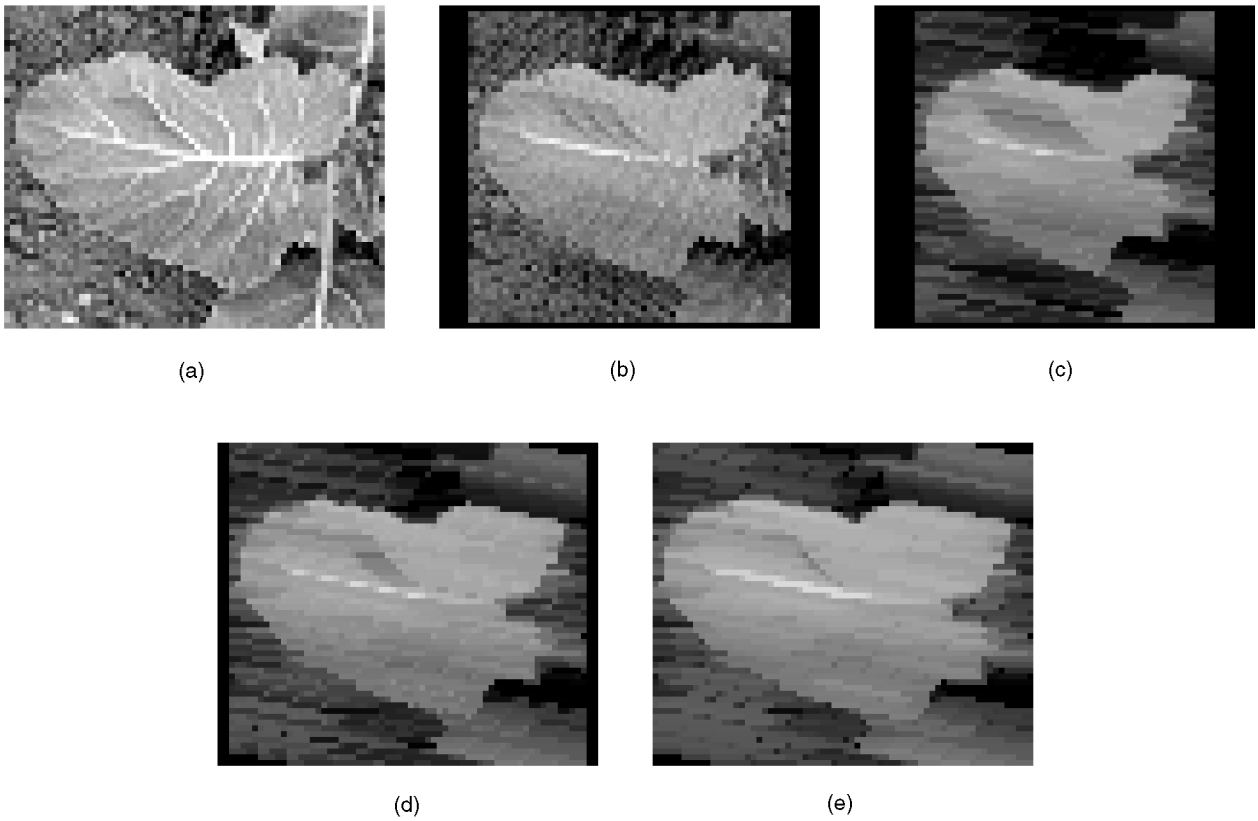


Fig. 3. Recursive TI opening by $L_{15_4,(5,-1)}$ (i.e., ▬▬▬) of the image shown in Fig. 2a: Successive steps based on the decomposition of erosions and dilations into periodic and plain SEs ((4) and (5)). (a) Input image. (b) Erosion of (a) by $P_{3,(5,-1)}$. (c) Erosion of (b) by $L_{5_4,(5,-1)}$. (d) Dilation of (c) by $\check{P}_{3,(5,-1)}$. (e) Dilation of (d) by $\check{L}_{5_4,(5,-1)}$.

Bresenham lines, it can be observed that the following equation applies:

$$\forall\, i \in\{1,\ldots,k\},\quad \gamma_{BL_{\lambda_i},(dx,dy)}(f)(x) \in \{\gamma_{L_{\lambda_1},(dx,dy)}(f)(x),\ldots,\gamma_{L_{\lambda_k},(dx,dy)}(f)(x)\}.$$

It follows that the union (i.e., point-wise maximum $\vee$ for discrete gray-scale images) of all k possible non-TI openings is identical to the union of the openings by the k possible line segments (the same result holds for closings ϕ by replacing the point-wise maximum with the point-wise minimum $\wedge$):

$$\bigvee_{i=1}^{i=k}\gamma_{BL_{\lambda_i},(dx,dy)} = \bigvee_{i=1}^{i=k}\gamma_{L_{\lambda_i},(dx,dy),} \tag{6}$$

$$\bigwedge_{i=1}^{i=k}\phi_{BL_{\lambda_i},(dx,dy)} = \bigwedge_{i=1}^{i=k}\phi_{L_{\lambda_i},(dx,dy).} \tag{7}$$

This is illustrated in Fig. 4. Figs. 4b, 4c, 4d, 4e, and 4f show the TI openings with the five SEs defined along a Bresenham line of slope $-1/5$. Their union shown in Fig. 4g is identical to the union of the five non-TI openings shown in Figs. 2b, 2c, 2d, 2e, and 2f (see Fig. 4h).

Matheron [30] demonstrated that any union of openings is itself an opening in the sense that it shares all the algebraic properties of a morphological opening (i.e., idempotence, increasingness, and antiextensivity). The union of openings (6) is therefore an algebraic opening (the same applies for the intersection of closings by duality with respect to complementation). Note that contrary to the recursive TI implementation of erosions and dilations ((4) and (5)), the length λ in (6) and (7) does not need to be a multiple of k. Regarding computational cost, the union of non-TI openings (right term in (6)) requires $6k$ min/max comparisons per pixel. This is more expensive than the $2k+4$ comparisons required for a given $L_{\lambda_i,(dx,dy)}$ but it yields the union of the openings with all k SEs $L_{\lambda_i,(dx,dy)}$ directly.

3.3 Periodic Moving Histogram TI Algorithms

We first present the principle of this technique for rank filters (Section 3.3.1) and then show that it is at the basis of valuable morphological rank-based filters (Section 3.3.2).

3.3.1 Rank Filters

The recursive TI strategy for directional min/max filters (Section 3.2.1) concentrates on SEs whose length is a multiple of the periodicity defined by their slope. Min/max filters by line segment SEs of arbitrary lengths require another strategy stemming from the moving histogram technique, originally proposed in [24], for computing median filters in square windows and later extended to rank filters [31] and erosions/dilations by arbitrary shaped SEs [32].

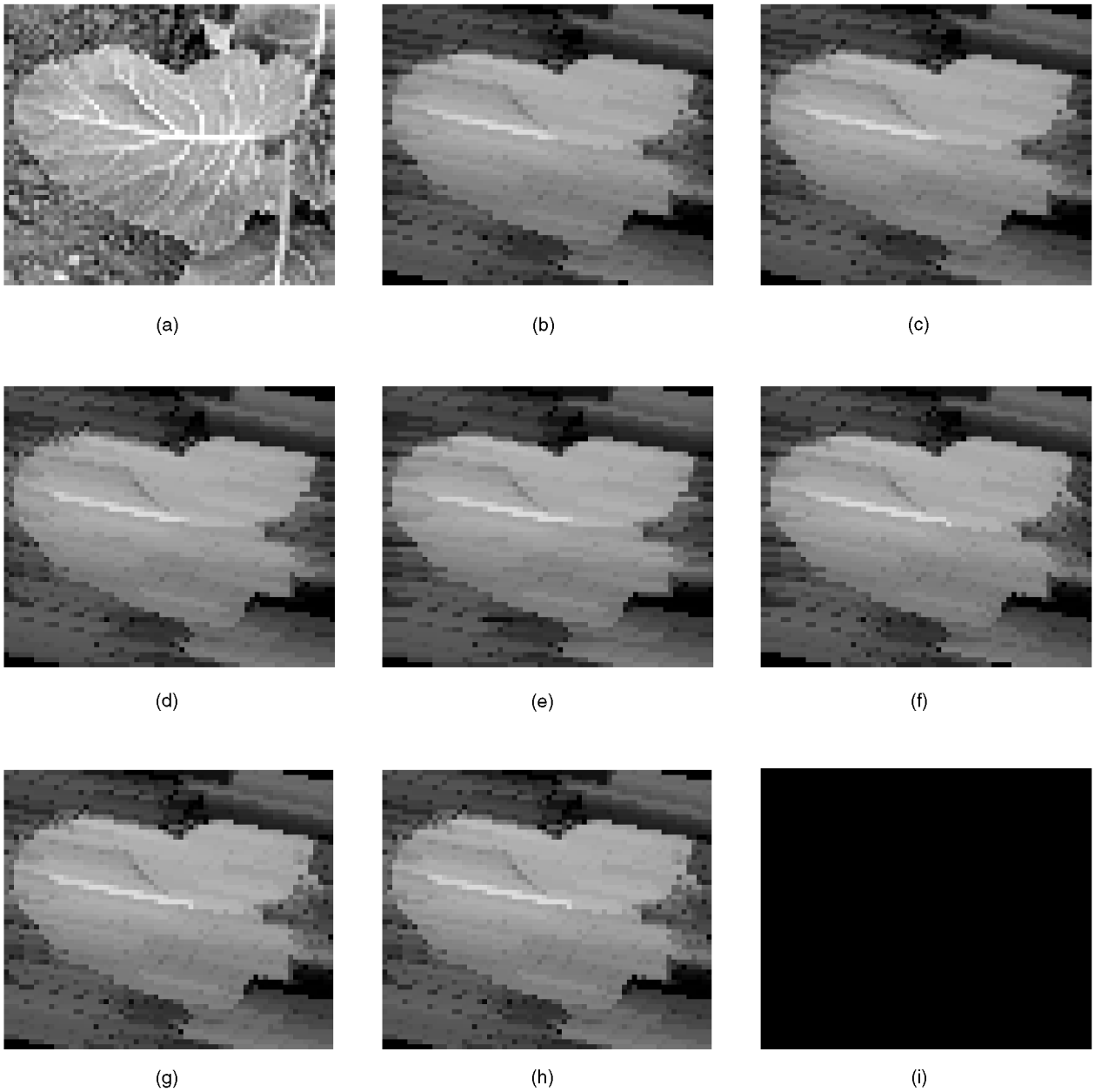


Fig. 4. (a) Input image f. Opening of the input image by the five possible SEs occurring along a discrete Bresenham line of slope $-1/5$: (b) Opening by $L_{15_1,(5,-1)}$: (▪▬▬▬▬), (c) Opening by $L_{15_2,(5,-1)}$: (▪▬▬▬▪), (d) opening by $L_{15_3,(5,-1)}$: (▪▬▬▪▪), (e) opening by $L_{15_4,(5,-1)}$: (▬▬▪▬), and (f) opening by $L_{15_5,(5,-1)}$: (▬▪▬▪). (g) Union of all five openings (i.e., Figs. 4b, 4c, 4d, 4e, and 4f): $\vee_{i=1}^{i=5}\gamma_{L_{15_i},(5,-1)}(f)$. (h) Union of the five non-TI openings shown in Figs. 2b, 2c, 2d, 2e, and 2f: $\vee_{i=1}^{i=5}\gamma_{BL_{15_i},(5,-1)}(f)$ is identical to the union of the five TI openings shown in (g), see (6). Therefore, the arithmetic difference between (g) and (h): $\vee_{i=1}^{i=5}\gamma_{L_{15_i},(5,-1)}(f) - \vee_{i=1}^{i=5}\gamma_{BL_{15_i},(5,-1)}(f)$ is a void image as illustrated in (i).

The key idea behind the moving histogram technique is that the frequency distribution (histogram) of the values of the pixels falling within a SE can be used to determine the minimum or maximum values of the image within this SE. When processing the next pixel, the histogram is updated[3] by taking into account the pixels that come out of the SE and those that come in. For example, in the trivial case of a horizontal line segment, there is only one pixel coming in and out whatever the length of the line segment when translating it along the horizontal direction.

As pointed out in [27, p. 82], the scanning order of the image pixels should be chosen so as to minimize the number of pixels coming in and out. For a SE in the form of a connected line segment $L_{\lambda,(dx,dy)}$ of arbitrary slope dy/dx and length λ, we introduce the notion of a *periodic image scan*.

Given a Euclidean line of slope dy/dx progressively sweeping the image plane, the periodic image scan consists of scanning the image pixels falling on the Euclidean line each time it intersects image pixels. By doing so, there are

3. Not only is the histogram updated, but the values of the pixels coming in and out are checked to see whether they are greater or lower than the previously computed min/max (or arbitrary rank) values so as to avoid sorting the whole histogram again to find the new min/max (or arbitrary rank) values.

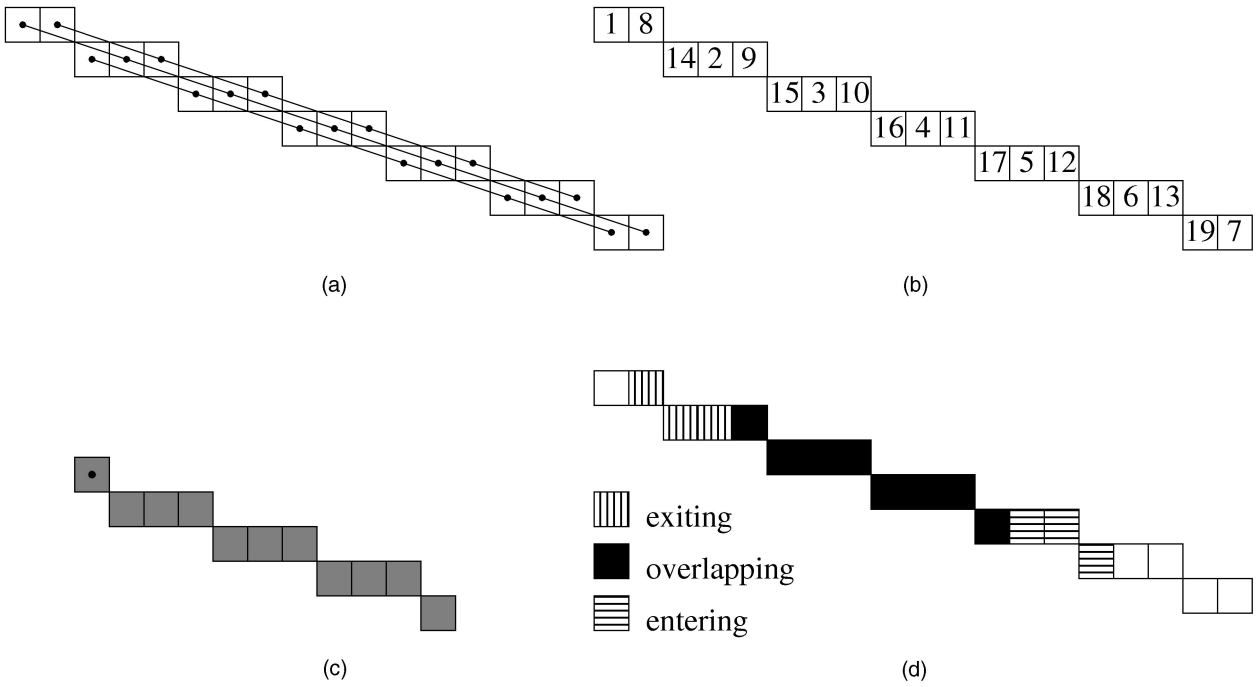


Fig. 5. Periodic moving histogram scheme. (a) A discrete Bresenham line of slope $-1/3$ and the three Euclidean lines sweeping all its pixels. (b) The periodic scanning order. (c) The connected line SE $L_{11_2,(3,-1)}$, its origin being marked by the black disk. (d) SE sliding from index 8 to index 9. (See text for a detailed explanation).

exactly k pixels coming in and out between two successive scanned pixels whatever the length of the SE. In practice, the periodic scan can be embedded within the sweeping of the image plane by a discrete Bresenham line [9]. Indeed, it can be observed [26] that a Bresenham line is itself exactly swept by k Euclidean lines of slope dy/dx.

An example is detailed in Fig. 5. Fig. 5a represents a discrete Bresenham line of slope $-1/3$ and the three Euclidean lines sweeping all its pixels. Note that each two successive pixels falling on a given Euclidean line are $k = 3$ pixels apart. Fig. 5b shows the corresponding periodic scanning order. The histogram is initialized at the start of each Euclidean line, i.e., indices 1, 8, and 14. In Fig. 5c, a connected line SE defined over the Bresenham line is displayed, namely, $L_{11_2,(3,-1)}$ (its origin being marked by the black disk). Fig. 5d shows this SE sliding from index 8 to index 9: Black pixels are common to each position, the three pixels hatched with vertical lines are coming out, and those hatched with horizontal lines are coming in. Notice that the number of pixels coming in and out equals k whatever the length of the SE. This figure also illustrates that the periodic scanning maximizes the overlapping degree between two successive positions of the SE. For example, a forward scan of the image pixels leads to five pixels coming in and out. Moreover, this number increases with the length of the SE. In general, for a SE length in the form of a multiple n of the periodicity k, there would be either $n\min(|dx|, |dy|)$ or $n\min(|dx|, |dy|) + 1$ pixels coming in and out for the scanning order maximizing the number of overlapping pixels, either line-wise or column-wise, depending on the slope of the SE and excluding the trivial case where the SE slope equals one of the principal directions of the grid.

Finally, notice that the total number of Euclidean lines necessary for sweeping the image plane is equal to $k(inc + cst + dcr)$, where inc, cst, and dcr refer to the length of the increasing, constant, and decreasing regions displayed in [9, Fig. 2, p. 463]. Consequently, the histogram must be initialized $k(inc + cst + dcr)$ times.

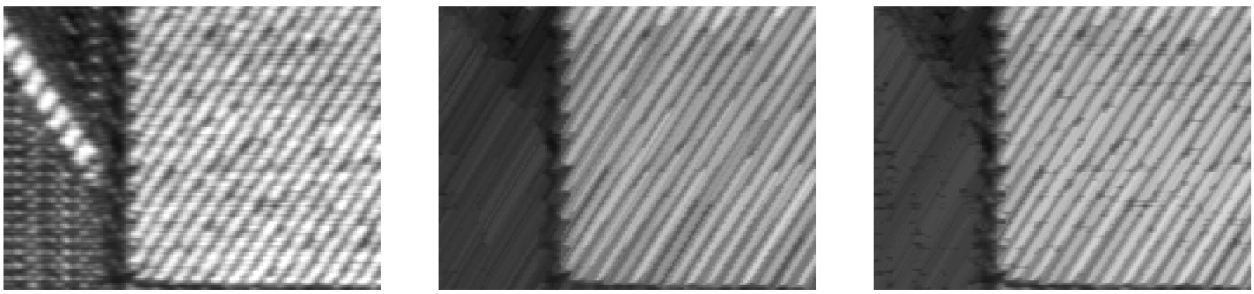
3.3.2 Rank-Based Morphological Filters

Apart from its suitability for line segment SEs of arbitrary length, the moving histogram strategy has the advantage of being designed for the computation of rank filters [33] of arbitrary rank rather than the sole min and max ranks as for the recursive TI strategy developed in Section 3.2. Besides, rank filters are themselves the basis of very useful, yet little known, morphological filters [34], [35] called rank-max openings [36], [37].

Given an arbitrary flat structuring element B whose cardinality $\mathrm{card}(B)$ equals n pixels, the rank-max opening of parameter r is equivalent to the union of the morphological openings by the structuring elements B_i included in B and containing r pixels. We denote by $\gamma_{B,r}$ the resulting algebraic opening:

$$\gamma_{B,r} = \bigvee_i \{\gamma_{B_i} \mid B_i \subseteq B \wedge \mathrm{card}(B_i) = r\}, \tag{8}$$

where $1 \leq r \leq n = \mathrm{card}(B)$. Notice that the smaller r is, the less active is the corresponding rank-max opening. More precisely, the following ordering relationship holds: $\gamma_{B,n} = \gamma_B \leq \gamma_{B,n-1} \leq \cdots \leq \gamma_{B,2} \leq \gamma_{B,1} = I$, where I denotes the identity transform. Fig. 6 shows the output of both a rank-max directional opening and the corresponding morphological opening. As expected from (6), the rank-max opening is less sensitive to the presence of small gaps



(a) (b) (c)

Fig. 6. Comparison between the output of a rank-max (parametric) directional opening and the corresponding morphological opening. (a) Input image sample f from a woolen fabric. (b) Morphological opening by the SE $L_{35_1,(5,8)}$ which corresponds to the rank-max opening with r set to the cardinal number of the SE, i.e., 35: $\gamma_{L_{35_1,(5,8)},35}(f) = \gamma_{L_{35_1,(5,8)}}(f)$. (c) Rank-max opening with r set to 32, i.e., allowing for gaps of up to three pixels: $\gamma_{L_{35_1,(5,8)},32}(f)$.

in the corresponding direction (gaps up to $n-r$ pixels are allowed along the considered direction).

From a computational point of view, it can be seen that the upper bound for the number of distinct SEs B_i in (8) equals $n!/r!(n-r)!$. This number is too large for most real applications. Fortunately, it can be shown [38, pp. 201-202] that the rank-max opening is equivalent to the intersection between the input image and the dilation by $\check{B}$ of its rank filter ξ using B as the mask and $n-r+1$ as the rank:[4]

$$\gamma_{B,r} = I \wedge \delta_{\check{B}} \xi_{B,n-r+1}. \tag{9}$$

Hence, a TI implementation of a directional rank-max opening by an arbitrary line segment $L_{\lambda,(dx,dy)}$ and parameter r can be implemented at the cost of two rank TI directional filters detailed in Section 3.3.1: The first being for the rank $\lambda - r + 1$ and the second for the rank n, i.e., a dilation:

$$\gamma_{L_{\lambda,(dx,dy)},r} = I \ \wedge \ \xi_{\check{L}_{\lambda,(dx,dy)},\lambda} \, \xi_{L_{\lambda,(dx,dy)},\lambda-r+1}. \tag{10}$$

By duality with respect to complementation, the following equation applies to rank-min (or simply parametric) closings:

$$\phi_{L_{\lambda,(dx,dy)},r} = I \ \vee \ \xi_{\check{L}_{\lambda,(dx,dy)},1} \, \xi_{L_{\lambda,(dx,dy)},r}. \tag{11}$$

A detailed discussion about rank-max filters and their relationships with other filters is provided in [40]. The practical interest of rank-max openings (and the dual rank-min closings) is illustrated in [36] for the processing of digitised X-ray angiographic images. More recently, rank-max openings with line segments have been used in [41], [42] for the extraction of the laid and chain lines in paper watermarks and [12] for the detection of the local orientation of thin elongated objects in noisy images using a bank of rank-max opening and rank-min closing filters. With the concept of periodic image scan introduced in Section 3.3, an efficient translation invariant implementation of directional rank-max openings (10) and rank-min closings (11) is available. Real applications are presented in Section 5.

4. Wilson [39] independently introduced the related concept of *shape inference* transform which is defined as a rank filter followed by a dilation with the reflected SE (i.e., the intersection between the identity transform and the shape inference transform leads to the rank-max opening).

4 Performance Evaluation

In this section, we compare algorithms for computing morphological or rank filters with line segments in arbitrary directions. We consider three criteria: computational complexity (given an image of fixed size), memory usage, and translation invariance.

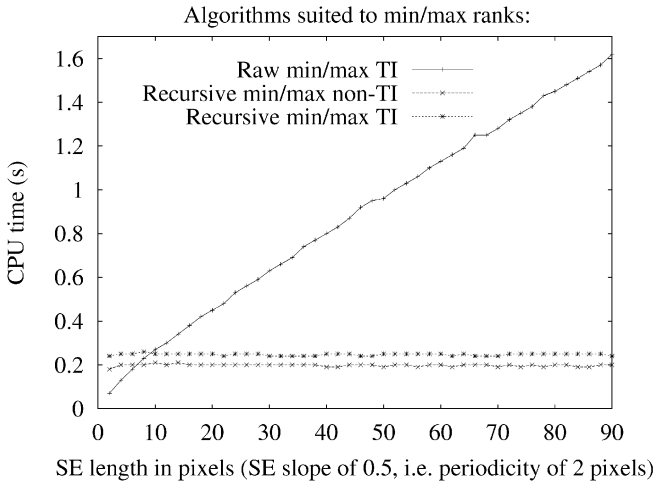
- **Raw TI min/max filter**. The complexity is $O(\lambda)$, where λ denotes the number of pixels of the line segment ($\lambda - 1$ min/max comparisons per pixel for a min/max filter). An extra image having the same size as the original image is required for the output image.
- **Recursive non-TI min/max filter [9]**. The complexity is $O(1)$ (three min/max comparisons per pixel for a min/max filter). All computations can be done in place, i.e., no extra image is necessary for storing the results (just two extra line buffers are necessary, see [9]).
- **Recursive TI min/max filter**. The complexity is also in $O(1)$ but the number of min/max comparisons depends on the slope of the line segment (this number equals $2+k$ comparisons per pixel for a min/max filter). The min/max with the periodic SE $P_{n,(dx,dy)}$ can be performed in place and the following min/max filter with the plain SE $L_{k_i,(dx,dy)}$ requires an extra image for storing the final result (3). Alternatively, TI openings or closings can also be obtained by unioning or intersecting k recursive non-TI openings or closings. The complexity is still $O(1)$ but the number of min/max comparisons per pixel equals $6k$ for the resulting TI algebraic opening (6) or closing (7).
- **Raw TI rank filter**. The complexity is $O(\lambda)$, where λ denotes the number of pixels of the line segment. Indeed, since discrete values in a restricted range need to be sorted, the distributive sorting technique [43] which has $O(n)$ complexity, where n denotes the number of elements to sort, can be considered. One extra image is required for storing the result.
- **Moving histogram along Bresenham lines rank filter.** The complexity of this non-TI implementation is $O(1)$ since there is one pixel coming in and out whatever the length of the Bresenham line segment (we neglect the $inc + cst + dcr$ sorting steps for

initializing the histogram arrays). In addition, the total number of operations is independent of the chosen slope. The processing time is data dependent as already reported in [24] for the computation of the median within a square window using a moving histogram. However, this influence is negligible in practice.

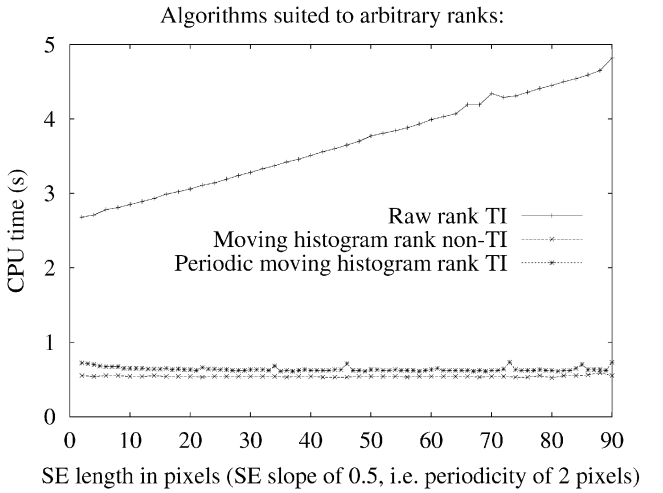
- **Periodic moving histogram TI rank filter**. The complexity is $O(1)$, neglecting the $k(inc + cst + dcr)$ sorting steps for initializing the histogram arrays. As with the recursive TI min/max filters, the computation time increases with k: There are k pixels coming in and out between two successive positions of the periodic image scan. An extra image for storing the results is required. The number of histogram bins depends on the number of gray-scale values of the input image. As for the moving histogram non-TI implementation, the processing time is slightly data dependent.

In Fig. 7, a plot of CPU time versus SE length highlights the computational complexity of all six algorithms. The processed image is a $1,024 \times 1,024$ input gray-scale image representing a two-fractional Brownian motion of dimension 2.5 generated by the Fourier filtering method implementing the pseudocode given in [44, p. 108] and with a seed value equal to 1. The generated image has been converted to unsigned char data type with values in the range $[0, 254]$ leaving the value 255 free for handling image borders in the moving histogram implementation. Table 1 summarizes the merits and drawbacks of each algorithm. In practice, the implementation choice must be driven by the application constraints (trade-off between speed and translation invariance). Also, it is important to realize that, for small SE lengths, raw algorithms (i.e., $O(\lambda)$ complexity) may be faster than the corresponding algorithms with $O(1)$ complexity, the break-even point depending on the slope of the SE (since it defines the periodicity k). When determining the break-even points for two given implementations, one should not forget that the length of a SE must be greater than k because a discrete slope dx/dy is defined for line segments having at least $k + 1$ pixels ($k = \max(|dx|, |dy|)$). In practice, we consider a SE length equal to at least twice the periodicity of the underlying discrete line. For example, Fig. 8 allows us to determine the break-even points between raw and periodic moving histogram rank algorithms for increasing values of k. For the raw TI algorithm, plots have been produced for SE lengths equal to nk, $n \in \{2, 3, 4\}$. Note that for the algorithms based on moving histograms, the CPU time is independent of the length of the SE. Therefore, the break-even point moves in favor of the periodic moving histogram algorithm when the length of the SE increases. The moving histogram *non-TI* plot has also been included to highlight that neither the periodicity nor the length of the SE have an effect on the CPU time.

Finally, it is worth noting that the number of possible slopes is finite but can be very large, the angular resolution increasing with the length of the SE. More precisely, the total number of distinct discrete slopes for dx and dy in the integer range $[-i, i]$ is given by $4\,\mathrm{card}(F_i) - 4$, where F_i is the Farey sequence[5] [45] of order i.



(a)



(b)

Fig. 7. Plot of CPU time versus length of the SE for a dilation with a slope of $1/2$ and a $1,024 \times 1,024$ input gray-scale image of a fractional Brownian motion with integer values in the range $[0, 254]$: comparison between (top) raw TI, recursive non-TI, and recursive TI for min/max filters, and (bottom) raw TI, moving histogram non-TI, and periodic moving histogram TI rank filters. The programs were written in ANSI C and run on an Intel P-II 450 MHz personal computer.

5 Applications

We first emphasize the use of directional openings and closings for image filtering and segmentation (Section 5.1). We then concentrate on issues related to directional granulometries (Section 5.2). Finally, we propose and develop the concept of morphological orientation field which allows for the computation of the local orientation of the image structures at each pixel (Section 5.3).

5. The Farey sequence F_i of order $i \geq 1$ is the ascending sequence of all fractions p/q for which $0 \leq p/q \leq 1$, $q \leq i$, and p and q are nonnegative integers with no common divisors other than 1.

TABLE 1
Summary Table of Algorithms for Computing Min/Max Filters with a Line Segment of Length λ (in Pixels) of Slope in the Form of the Irreducible Fraction dy/dx and Periodicity $k = \max(|dx|, |dy|)$

	Raw	Recursive [9]	Recursive TI (Sec. III-B.1)	Moving hist. along Bres. lines	Periodic moving hist. (Sec. III-B.2)
Translation invariance	yes	no	yes	no	yes
Length λ of SE	any	any	multiple of k	any	any
Computational complexity	$O(\lambda)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
Comparisons/pixel	$\lambda - 1$	3	$2 + k$	data dependent[a]	data dependent[b]
Image data type	any	any	any	integer	integer
Memory usage	2	1	2	1	2
Suited to rank filters	yes	no	no	yes	yes

The raw algorithm refers to both min/max and rank filters. The memory usage is in number of image buffers necessary to run the algorithm.
[a] *2 pixels to process for updating histogram.*
[b] $2k$ *pixels to process for updating histogram.*

5.1 Filtering and Segmentation

We believe that the proposed directional filters are useful in a wide range of practical applications such as fingerprint and fiber analysis, document interpretation (where thin lines have to be tracked), industrial inspection (where cracks, scratches, and other elongated defects have to be detected), and satellite imagery for the detection of road networks and lineaments. Low-level image filtering tasks by directional morphological filters are illustrated in Fig. 9 on a binary image of fibers acquired by an electron probe microanalyzer. Fig. 9 shows that the union of directional TI openings of rank five by all discrete line segments of 15 pixels allows us to remove the noise and nonfibre objects while preserving even those fibers containing gaps of up to four pixels (and having at least a length of 11 pixels). A TI implementation is important because fibers are so thin that the output of the non-TI implementation varies significantly when shifting the image definition domain.

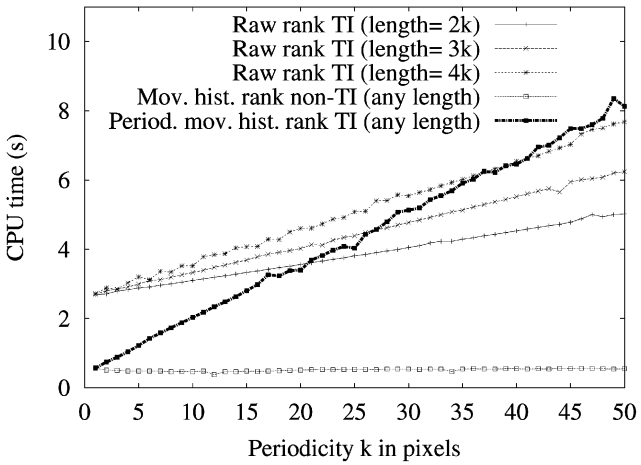


Fig. 8. Break-even points between raw and periodic moving histograms rank TI algorithms depending on the periodicity k of the underlying line segment. The moving histogram *non-TI* algorithm plot has been included for comparison purposes. The input $1,024 \times 1,024$ image is identical to that used for producing Fig. 7.

A bank of directional openings and/or closings can also be used for extracting long thin objects in an image. This approach is illustrated in Fig. 10 for the extraction of bus-like structures in a one meter resolution satellite image. The first step of the methodology consists in filtering the input image by removing all connected image structures smaller than a bus. Assuming that a bus is a linear object of at least 12 pixels, this filtering is achieved by performing a union of openings with all possible directional structuring elements of length equal to 11 pixels. The output of this filter is used as a marker image for a morphological reconstruction by dilation of the input image (Fig. 10b). The same filter, but with all possible directional structuring elements slightly longer than buses (i.e., 15 pixels) is then considered (Fig. 10c). By performing the difference between the second and first filter, we obtain a strong response in regions where linear structures in the range size of buses are present (Fig. 10d). A mask of the detected buses is then obtained by thresholding this image (Fig. 10e).

In addition to low-level directional image filtering tasks, the proposed TI implementations allow for the computation of directional granulometries and orientation fields. These two higher-level image analysis tasks are detailed hereafter.

5.2 Directional Granulometries

The concept of granulometry [46], [47] is analogous to the sifting of grains through screens of increasing mesh size. At each step, the grains remaining in the sieve are those that are larger than the mesh. The granulometry is depicted as a diagram which plots the amount of remaining grains versus the mesh size. The discrete derivative of this diagram is often referred to as the pattern spectrum [48].

In mathematical terms, a granulometry is defined by a transformation Γ having a size parameter λ and satisfying the antiextensivity ($\Gamma \leq I$), increasingness ($f \leq g \Rightarrow \Gamma(f) \leq \Gamma(g)$), and absorption properties

$$\left(\Gamma_{\lambda_1}\Gamma_{\lambda_2} = \Gamma_{\lambda_2}\Gamma_{\lambda_2} = \Gamma_{\max(\lambda_1,\lambda_2)}\right).$$

(a) (b)

Fig. 9. Filtering of images of fibers corrupted by electronic noise due to the acquisition device (microprobe). A union of directional rank-max openings removes the noise and preserves most fibers. (a) 295×164 image f of fibers. (b) Output image: $\vee_\theta \gamma_{L_{15_1},\theta,11}(f)$.

Note that, for $\lambda_1 = \lambda_2$, the idempotence property is a particular case of the absorption property. By definition, the idempotence, antiextensivity, and increasingness properties characterize an opening transform. However, not all openings with SEs of increasing size satisfy the absorption property. For instance, it is not fulfilled when considering line segments of increasing length (one pixel at a time) in an arbitrary direction, even when considering the raw TI approach. Indeed, the increase in length must occur k pixels at a time, as already pointed out in [28]. However, by considering the union of openings described in Section 3.2, one can proceed one pixel at a time while satisfying all axioms of a granulometry. The resulting directional granulometry Γ for a given slope dy/dx and size parameter λ is then defined as follows:

$$\Gamma_{\lambda,(dx,dy)} = \bigvee_{i=1}^{i=k} \gamma_{L_{\lambda_i,(dx,dy)}}. \tag{12}$$

Early work concerning morphological directional granulometries for texture classification is due to [49]. Indeed, they performed openings/closings with line segments of variable length and along the principal directions of the square grid. The texture signature was then defined as the sum of the gray levels of the transformed image. Equation (12) and the proposed TI algorithms allow for a generalization of the procedure to arbitrary directions.

Note that, when considering rank-max directional openings for building up granulometries, one has to be very careful. Indeed, it can be shown that the absorption property of a granulometry is not satisfied when using partial line segments containing a fixed percentage of the full line segment. However, when using the same r value for the family of openings of increasing size, all axioms of a granulometry are satisfied. The other side of the coin is that the sensitivity to noise then increases with the length λ of the SE. This may hamper the sound interpretation of the resulting granulometry.

5.3 Orientation Field

We propose to define the orientation at a given pixel x in an image as the orientation of the line segment that minimizes the difference between the gray-level value in the original image at x and the gray-level value at the same location in the image filtered by the considered line segment. Therefore, we are interested in the local orientation of an elongated image structure rather than the local direction of the vector gradient as produced by Sobel-like operators. Openings should be used for image structures that are brighter than their background (i.e., "positive" image structures) and closings for image structures darker than their background (i.e., "negative" image structures). In mathematical terms, we define the positive orientation at a given image pixel x and for a given scale λ as the orientation of the directional morphological opening of length λ which modifies the least original image value at position x. We denote the positive orientation by Dir^+, the negative orientation Dir^- being defined by duality:

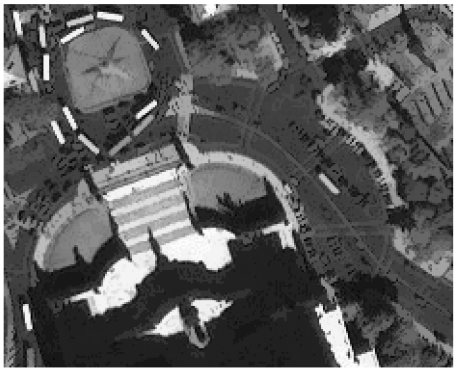
$$Dir^+_\lambda(f)(x) = \{\theta_i \mid \gamma_{L_{\lambda,\theta_i}}(f)(x) \geq \gamma_{L_{\lambda,\theta_j}}(f)(x),\ \forall\, \theta_i \neq \theta_j\}, \tag{13}$$

$$Dir^-_\lambda(f)(x) = \{\theta_i \mid \phi_{L_{\lambda,\theta_i}}(f)(x) \leq \phi_{L_{\lambda,\theta_j}}(f)(x),\ \forall\, \theta_i \neq \theta_j\}. \tag{14}$$

If all orientations output the same value, it means that there is no orientation for the chosen structuring length. An arbitrary orientation can then be selected because the strength of the orientation (see below) will be zero. However, for pixels belonging to wide (with respect to the length λ of the structuring element) oriented objects of constant gray tone, several distinct orientations may output the maximum value (we concentrate here on orientation by openings Dir^+; similar developments apply for the orientation by closings Dir^-). Assuming that the range of orientations at the maximum value is connected, we break the tie by selecting the middle orientation. If the range is not connected, it means that there are oriented structures crossing each other. We then choose the main orientation as the orientation of the middle of one of the connected range of orientations producing the maximum value. If required by the application under study, rather than selecting a unique orientation, the positive (respectively, negative) directional signature at a given pixel can be analyzed by plotting the normalized opened (respectively, closed) values versus the orientation of the line segment. This signature can then be used to detect crossing lines, flat zones, etc.

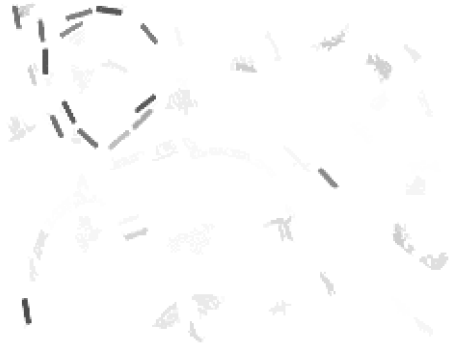
We also define the following quantities for each point x of the input image f:

(a)



(b)

(c)



(d)

(e)

Fig. 10. Extraction of bus-like structures appearing in a one meter resolution satellite image (IKONOS) using union of openings by line segments. (a) Input satellite image (IKONOS) showing Piazza Venezia in Rome. (b) Union of openings by line segments slightly shorter than buses (followed by reconstruction by dilation of original image). (c) Union of openings by line segments slightly longer than buses (followed by reconstruction by dilation of original image). (d) Difference between image (b) and (c). (e) Global threshold of (d).

$$Max_\lambda^+(f)(x) = \{\gamma_{L_{\lambda,\theta_i}}(f)(x) \mid \gamma_{L_{\lambda,\theta_i}}(f)(x) \geq \gamma_{L_{\lambda,\theta_j}}(f)(x),\ \forall\ \theta_i \neq \theta_j\}, \quad (15)$$

$$Min_\lambda^+(f)(x) = \{\gamma_{L_{\lambda,\theta_i}}(f)(x) \mid \gamma_{L_{\lambda,\theta_i}}(f)(x) \leq \gamma_{L_{\lambda,\theta_j,\lambda}}(f)(x),\ \forall\ \theta_i \neq \theta_j\}, \quad (16)$$

$$Gdir_\lambda^+(f)(x) = Max_\lambda^+(f)(x) - Min_\lambda^+(f)(x). \quad (17)$$

$Gdir_\lambda^+$ can be interpreted as the strength of the positive orientation: It will output a small value if there is no predominant orientation for a structuring element length of λ pixels. $Max_\lambda^-, Min_\lambda^-$, and $Gdir_\lambda^-$ are defined by replacing the opening with the closing in (15) and (16).

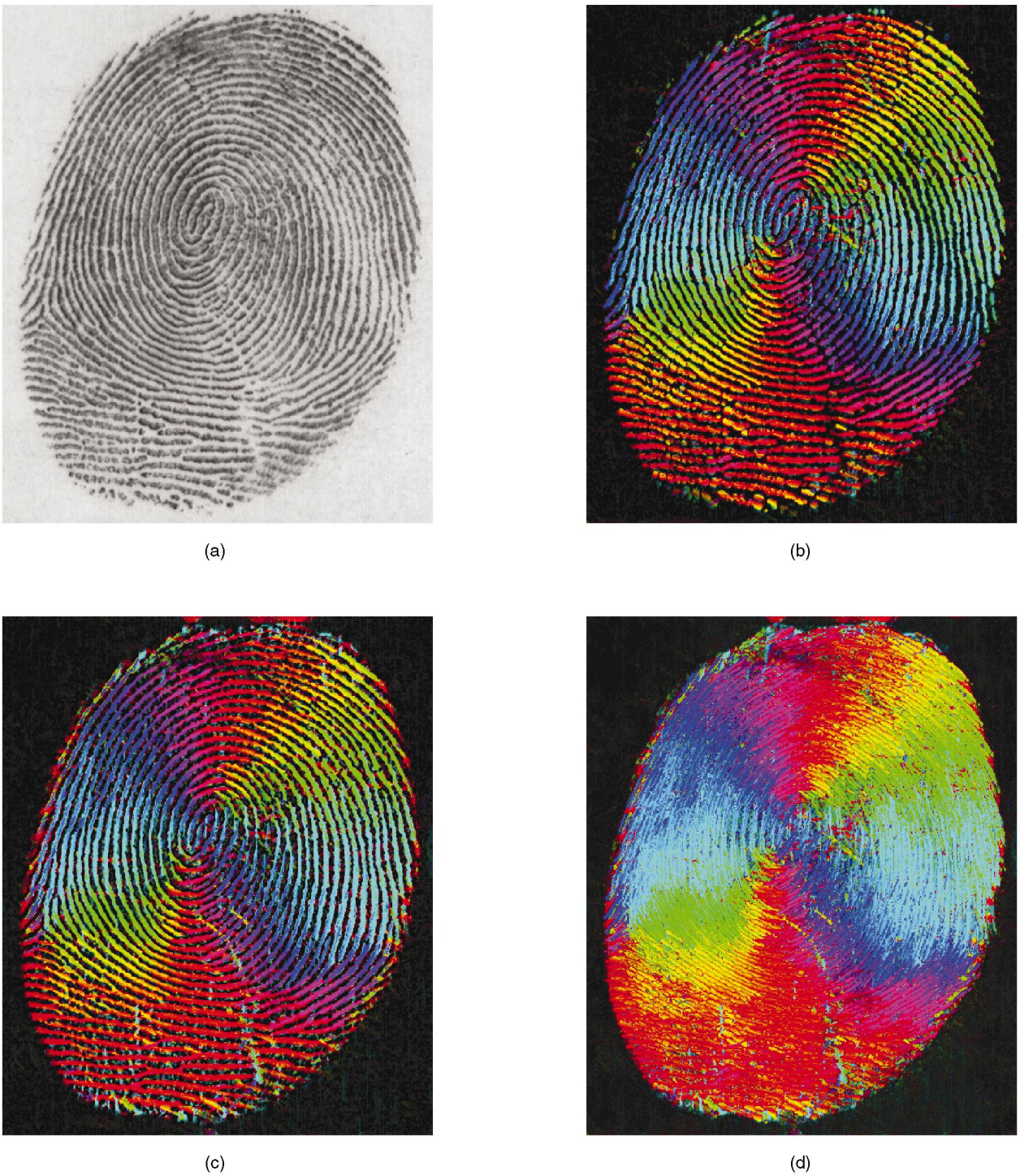


Fig. 11. (a) An Input fingerprint f. (b) $Dir^{+}(f)$: Local orientation using openings (13), i.e., orientation of bright ridges. (c) $Dir^{-}(f)$: Local orientation using closings (13), i.e., orientation of dark ridges. (d) $Dir(f)$: Combination of both transformations (18) showing the local orientation of both dark and bright oriented structures. The SE used in this experiment is a line segment of 21 pixels using the moving histogram TI implementation and a rank of five for the corresponding rank-max openings and closings (i.e., r value of 17).

When comparing the values of $Gdir^{+}$ and $Gdir^{-}$ of a given pixel, it is possible to detect whether it belongs to a positive or negative image structure: positive, if $Gdir^{+} > Gdir^{-}$, negative, otherwise. We denote by $Gdir$ the point-wise maximum between the images $Gdir^{+}$ and $Gdir^{-}$:

$$Gdir = Gdir^{+} \vee Gdir^{-}.$$

The image of directions Dir is then defined as follows:

$$Dir_{\lambda}(f)(x) = \begin{cases} Dir^{+}_{\lambda}(f)(x), & \text{if } Gdir_{\lambda}(f)(x) = Gdir^{+}_{\lambda}(f)(x), \\ Dir^{-}_{\lambda}(f)(x), & \text{otherwise.} \end{cases} \tag{18}$$

A color representation of the local orientation information is then simply achieved by equating the orientation

(a) (b) (c)

Fig. 12. Fiber orientation of a noisy image using directional openings: plain versus rank-max directional openings. (a) 300×300 scanning electron microprobe image of fibers. (b) Orientation field with directional plain openings: $\gamma_{L_{25_1,\theta}}$. (c) Orientation field with directional rank-max openings: $\gamma_{L_{25_1,\theta,21}}$.

information (i.e., either Dir^+, Dir^-, or Dir) to the hue and the strength of the orientation information (i.e., either $Gdir^+$, $Gdir^-$, or $Gdir$) to the lightness component of the image, the colors being fully saturated in all cases. For example, the color representations of all three types of local orientation are illustrated in Fig. 11 on a fingerprint. A comparative study with other techniques for computing directional images of fingerprints such as the level curves [50] used in [51] is outside the scope of this study.

A last example illustrating the better performance of rank-max directional openings over plain directional openings in situations where the input image is affected by some type of noise is provided in Fig. 12. Fig. 12a displays a raw image of multiphase materials acquired by an electron probe microanalyzer. Since we are interested in measuring the orientation of the bright fibers, we can restrict the orientation analysis to the use of openings. Fig. 12b shows the output using plain openings, while Fig. 12c with rank-max openings using a value of 21 for r, i.e., a rank value of five for the rightmost rank filter in (10). The latter image provides more directional information because it allows for gaps created by electronic noise during image acquisition.

6 Extensions

So far, we have focused on algorithms and applications dealing with flat line segments. In this section, we show first that two-dimensional structuring elements can be obtained by cascading line segments. Second, we present some results concerning the extension of the recursive algorithms for gray-tone (nonflat) periodic line segments of constant gray slope.

6.1 Line Segment Cascades

Cascades leading to arbitrary oriented rectangles are first proposed. Then, we describe a method for generating discrete disks from cascades of periodic lines.

6.1.1 Leading to Arbitrary Rectangles

By cascading two erosions (respectively, dilations) with vertical and horizontal line segments, one achieves erosions (respectively, dilations) with square structuring elements:

$$\square_n = L_{n,(1,0)} \oplus L_{n,(0,1)}, \tag{19}$$

where $\square_n$ is a square of width n pixels. It has long been known that discrete diamond-shaped structuring elements cannot be generated by cascading erosions (respectively, dilations) with structuring elements at 45 and -45 degrees. Although logarithmic decompositions have been proposed [52] for speeding up operations with diamond-shaped sets, the following simple and efficient decomposition can be used instead:

$$\diamond_n = L_{n-1,(1,1)} \oplus L_{n-1,(1,-1)} \oplus \diamond_2, \tag{20}$$

where $\diamond_n$ is the diamond-shaped structuring element with a side of n pixels ($n \geq 2$), i.e., $\diamond_2$ is the four-connected neighborhood plus its central pixel. By definition, $\diamond_1$ is a single pixel. An example is shown in Fig. 13. All these decompositions substantially speed up operations for n large enough (6 (respectively, 10) min/max comparisons per pixel whatever the width n of the square (respectively, diamond) when using the recursive algorithm instead of $O(n^2)$ operations for the brute force algorithm). Note also that cascades of erosions/dilations by $\diamond_2$ currently used for

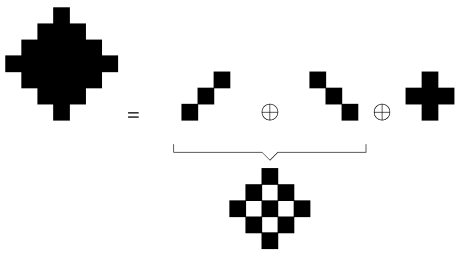


Fig. 13. Decomposition of a diamond-shaped structuring element with a width of four pixels using (20).

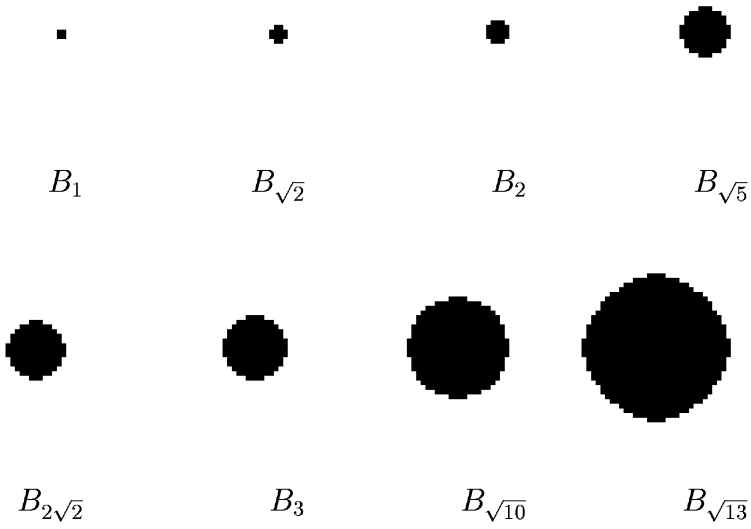


Fig. 14. Radial decompositions of discrete disks of increasing size using cascades of dilations with periodic lines. Each disk is symmetric, convex (see Section 3.1), and is opened by all disks of smaller size: $\gamma_{B_i}(B_j) = B_j$ for all $i \leq j$ (see text for the definition of each B_i).

generating operations with diamond-shaped structuring elements of larger size are not only less efficient than (20) but also only lead to diamonds of odd width.

More generally, a discrete rectangle Rec of side lengths in the form $ak+1$ and $bk+1$ and of arbitrary slope in the form of an irreducible fraction dx/dy can be generated by the following cascade:

$$Rec_{ak+1,bk+1,(dx,dy)} = L_{k_i,(dx,dy)} \oplus P_{a,(dx,dy)} \oplus L_{k_i,(-dy,dx)} \oplus P_{b,(-dy,dx)} \oplus \diamond_2, \quad (21)$$

where a and b are positive integers and $k = \max(|dx|, |dy|)$. In the following section, we concentrate on cascades starting from periodic lines and leading to discrete disks.

6.1.2 Leading to Discrete Disks

In Euclidean morphology, Matheron [30, p. 94] has shown that, for $\theta_1, \theta_2, \ldots \theta_n$ distinct in $[0, \pi)$, $k_1, k_2, \ldots, k_n > 0$, then $k_1 L_{\theta_1} \oplus k_2 L_{\theta_2} \oplus \cdots \oplus k_n L_{\theta_n}$ is a convex polygon of $2n$ sides whose opposite edges are of length $2k_i$ and have orientation given by θ_i. Adams [53] used this principle for generating disks (and spheres) of increasing size from cascades of dilations by discrete Bresenham line segments. In [28], it has been shown that cascades of periodic lines lead to better results in the sense that the resulting disks are symmetric. The approximation of a Euclidean disk (ball) B can be written as follows: $B \approx S_{\vec{\lambda},\vec{v}}$. However, no clue was given on how to select the vector of sizes and the corresponding vector of periodic lines. This can be achieved as follows: Suppose we look for the best discrete approximation of a Euclidean disk using cascades of line segments whose Euclidean length equals l. We then consider all distinct discrete slopes defined for this length and use the corresponding periodic lines with the maximal number of pixels so that the corresponding Euclidean line segment does not exceed l. We denote by B_l the corresponding approximation. For example, we have the following cascades for the five first approximations:

$$\begin{aligned}
B_1 &= P_{2,(1,0)} \oplus P_{2,(0,1)},\\
B_{\sqrt{2}} &= P_{2,(1,0)} \oplus P_{2,(0,1)} \oplus P_{2,(1,1)} \oplus P_{2,(1,-1)},\\
B_2 &= P_{3,(1,0)} \oplus P_{3,(0,1)} \oplus P_{2,(1,1)} \oplus P_{2,(1,-1)},\\
B_{\sqrt{5}} &= P_{3,(1,0)} \oplus P_{3,(0,1)} \oplus P_{2,(1,1)} \oplus P_{2,(1,-1)}\\
&\quad \oplus P_{2,(2,1)} \oplus P_{2,(2,-1)} \oplus P_{2,(1,2)} \oplus P_{2,(1,-2)},\\
B_{2\sqrt{2}} &= P_{3,(1,0)} \oplus P_{3,(0,1)} \oplus P_{3,(1,1)} \oplus P_{3,(1,-1)}\\
&\quad \oplus P_{2,(2,1)} \oplus P_{2,(2,-1)} \oplus P_{2,(1,2)} \oplus P_{2,(1,-2)}.
\end{aligned}$$

Fig. 14 displays the eight first disks. Notice that, by construction, these disks form a granulometry with size parameter given by l. In contrast, radial decompositions using Bresenham lines [53] cannot be used to generate a granulometric function or do the disks obtained by thresholding the Euclidean distance computed from a center pixel (the first element would be $\diamond_2$ and the second $\square_3$ but it is not opened by the first, i.e., the absorption property is not satisfied).

6.2 Algorithm for Nonflat Line Segments of Constant Gray-Tone Slope

The dilation of an image f with a gray-scale (also referred to as volumic or nonflat) structuring element B_v is denoted by $\delta_{B_v}(f)$ and is defined as follows for each point x:

$$[\delta_{B_v}(f)](x) = \max_{b \in B_v} \{f(x+b) + B_v(b)\}. \quad (22)$$

The erosion is defined by duality with respect to set complementation: $[\varepsilon_{B_v}(f)](x) = \min_{b \in B_v} \{f(x+b) - B_v(b)\}$.

Gray-scale structuring elements should be used with care because the corresponding erosions and dilations do not commute with scalings of the pixel intensity values [54],

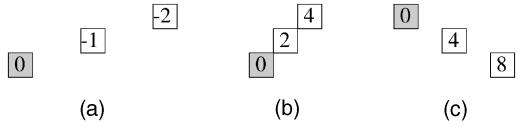


Fig. 15. Gray-scale periodic structuring elements. (a) $P_{3,(3,1),-1}$. (b) $P_{3,(1,1),2}$. (c) $P_{3,(2,-1),4}$.

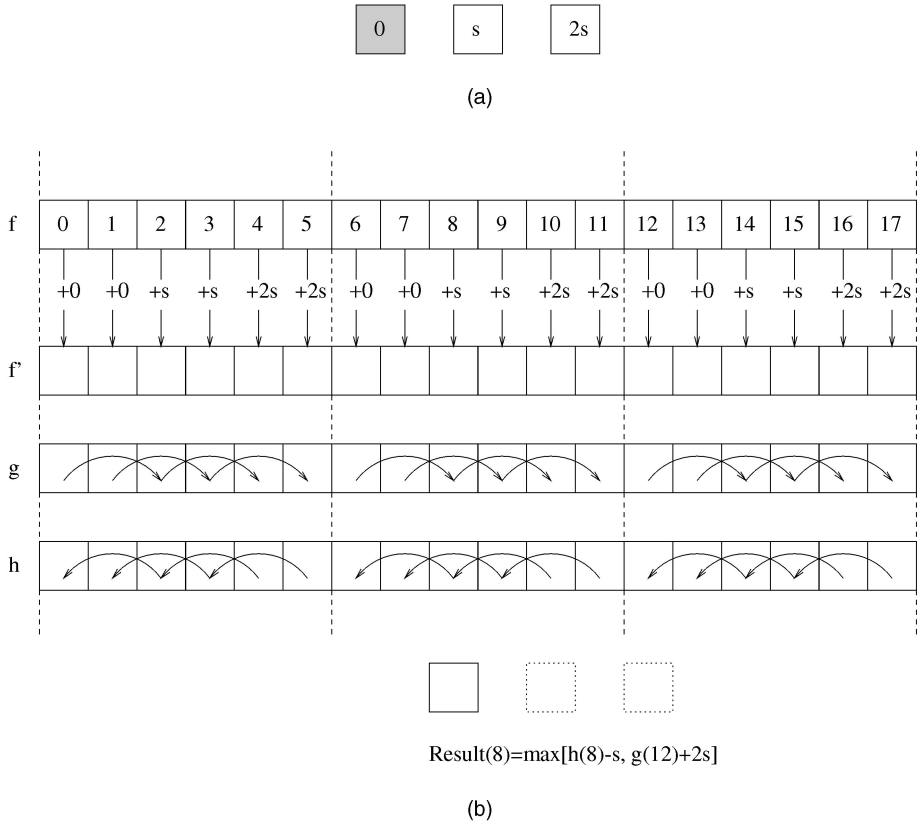


Fig. 16. Recursive min/max filter with a gray-scale periodic structuring element $P_{3,(2,0),s}$ (shaded origin). The structuring element is shown in (a) and the schematic of the algorithm in (b).

[55]. Nevertheless, these structuring elements are useful for some applications such as the rolling-ball algorithm and the computation of shadows of an image seen as a topographic surface [56].

We now show that the recursive procedure for computing min/max filters extends to gray-scale periodic structuring elements, i.e., structuring elements whose domain of definition is a periodic line and whose gray-scale values are defined as the index of the point i in the periodic line multiplied by a real number s defining the gray-scale slope: $P(i\vec{v}) = is,\ \forall i \in \{0, 1, \ldots, n-1\}$ [57], [58] (see also related developments in [59]). We denote by $P_{n,\vec{v},s}$ gray-scale periodic structuring elements of gray-scale slope s. Examples of gray-scale periodic structuring elements are presented in Fig. 15.

The algorithm requires an additional buffer f'. The values of the input image are copied in this buffer which is partitioned into blocks, as described in [9]. The periodic structuring element is then positioned at the first pixel of each block and the weights are added to each pixel of the block whose intersection with the structuring element is nonempty. The structuring element is then translated by one pixel to the right and the procedure is repeated until a weight has been added to all pixels of f' (there are $k-1$ translations per block). Once the values of the buffer f' have been calculated, the buffers g and h are computed from the buffer f' using the recursive procedure detailed in [9]. Finally, the resulting value at each position equals the maximum value between the value in the buffer h at the current position and the value in the buffer g at the current position plus $x + k(n-1)$. However, in order to ensure that the same weights are used for all positions, appropriate multiples of the slope s must be added to g and subtracted from h beforehand. For clarity and conciseness, we consider here the case of a *connected* periodic line of n pixels (i.e., the periodicity k is equal to 1) with the origin matching the first pixel of the periodic line (i.e., $o = 0$). In this situation, $x \bmod n$ times s must be removed from h and $(n-1) - (x+n-1) \bmod n$ times s must be added to g.

$$
\begin{aligned}
f'(x) &= f(x) + (x \bmod n)s \\
g(x) &= \begin{cases} f'(x) & \text{if } x \bmod n = 0, \\ \max[g(x-1), f'(x)] & \text{otherwise.} \end{cases} \\
h(x) &= \begin{cases} f'(x) & \text{if } (x+1) \bmod n = 0, \\ \max[h(x+1), f'(x)] & \text{otherwise.} \end{cases} \\
r(x) &= \max[g(x+n-1) + ((n-1) - (x+n-1) \bmod n)s, \\
&\qquad h(x) - (x \bmod n)s].
\end{aligned}
$$

This algorithm adapts directly to nonunitary periodicities (i.e., $k > 1$). An example is given in Fig. 16 for the periodic line $P_{3,(2,0),s}$. For conciseness, we use a structuring element of three pixels only. Since our algorithm requires three max comparisons per pixel whatever the number of pixels in the structuring element, speed gains are obtained for longer structuring elements. In practice, cascades of gray-scale line segments allow for the radial decomposition of the hemispheres utilized in the rolling ball algorithm as demonstrated in [53].

7 Conclusion and Perspectives

The primary contribution of this paper was to show that efficient algorithms for min/max and rank filters along line segments are possible without sacrificing the translation invariance property, while preserving the constant computational complexity of former non-TI algorithms. The resulting superior performance in terms of invariance to image translations is achieved at the cost of only a few extra computations directly proportional to the periodicity induced by the slope of the chosen SE.

Application examples have illustrated that the proposed algorithms will be of interest for a wide range of problems dealing with images containing oriented objects or textures and in situations where translation invariance cannot be compromised with speed. We are currently investigating the use of orientation fields for texture segmentation/identification and developing techniques for selecting the optimal parameter values in the proposed directional transformations. The choice of the r value in the rank morphological filters should be driven by the amount of noise corrupting the local connectivity of the oriented image patterns. The orientation resolution is constrained by the length of these patterns and can be automatically determined from a multiscale approach, the scale being defined by the length λ of the line segments. Directional granulometries and the corresponding pattern spectra will play a key role for choosing the optimal scale.

All developments presented in this paper can be extended to the processing of n-D images with line segments. The principles are identical but the implementation is more intricate especially with respect to the handling of the image boundaries. This issue has already been pointed out in [60] for the processing of 3D images along periodic lines.

Acknowledgments

P. Soille was with the Image Analysis Group of the BBSRC Silsoe Research Institute, Wrest Park, Silsoe MK 45 4DP, UK when this paper was first submitted.

References

[1] A. Dufour and C. Bonnet, "Influence of Unattended Features on the Processing of a Single Line Segment," *Spatial Vision,* vol. 9, no. 3, pp. 307-324, 1995.

[2] D. Kurylo, A. Reeves, and B. Scharf, "Expectancy of Line Segment Orientation," *Spatial Vision,* vol. 10, no. 2, pp. 149-162, 1996.

[3] J. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters," *J. Optical Soc. Am. A,* vol. 2, no. 7, pp. 1160-1169, July 1985.

[4] W.T. Freeman and E. H. Adelson, "The Design and Use of Steerable Filters," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 9, pp. 891-906, Sept. 1991.

[5] D. Jeulin and M. Kurdy, "Directional Mathematical Morphology for Oriented Image Restoration and Segmentation," *Acta Stereologica,* vol. 11, pp. 545-550, 1992.

[6] R. Bamberger and M. Smith, "A Filter Bank for the Directional Decomposition of Images: Theory and Design," *IEEE Trans. Signal Processing,* vol. 40, pp. 882-893, Apr. 1992.

[7] H. Yamada, K. Yamamoto, and K. Hosokawa, "Directional Mathematical Morphology and Reformalized Hough Transformation for the Analysis of Topographic Maps," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 4, pp. 380-387, Apr. 1993.

[8] E.R. Davies, "Vectorial Strategy for Designing Line Segment Detectors with High Orientation Strategy," *Electronic Letters,* vol. 33, no. 21, pp. 1774-1777, 1997.

[9] P. Soille, E. Breen, and R. Jones, "Recursive Implementation of Erosions and Dilations Along Discrete Lines at Arbitrary Angles," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 5, pp. 562-567, May 1996.

[10] S. Beucher, S. Kozyrev, and D. Gorokhovik, "Pré-Traitement Morphologique d'Images de Plis Postaux," *Colloque National sur l'Ecrit et le Document,* M. Gilloux, ed., pp. 133-140, Paris: CNED. 1996.

[11] R. Zwiggelaar and C. Taylor, "Abnormal Masses in Mammograms: Detection Using Scale-Orientation Signatures," *Proc. Medical Image Computing and Computer-Assisted Intervention—MICCAI '98,* W. Wells, A. Colchester, and S. Delp, eds., pp. 570-577, 1998.

[12] P. Soille and H. Talbot, "Image Structure Orientation Using Mathematical Morphology," *Proc. 14th Int'l Conf. Pattern Recognition,* A. Jain, S. Venkatesh, and B. Lovell, eds., vol. 2, pp. 1467-1469, Aug. 1998.

[13] P. Whelan, P. Soille, and A. Drimbarean, "Real-Time Registration of Paper Watermarks," *Real-Time Imaging,* vol. 7, no. 4, pp. 367-380, Aug. 2001.

[14] J. Bresenham, "Algorithm for Computer Control of Digital Plotter," *IBM System J.,* vol. 4, pp. 25-30, 1965.

[15] P. Soille, E. Breen, and R. Jones, "A Fast Algorithm for Min/Max Filters Along Lines of Arbitrary Orientation," *Proc. IEEE Workshop Nonlinear Signal and Image Processing,* I. Pitas, ed., pp. 987-990, June 1995.

[16] M. van Herk, "A Fast Algorithm for Local Minimum and Maximum Filters on Rectangular and Octogonal Kernels," *Pattern Recognition Letters,* vol. 13, pp. 517-521, 1992.

[17] J. Gil and M. Werman, "Computing 2D Min, Median, and Max Filters," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 5, pp. 504-507, May 1993.

[18] D. Gevorkian, J. Astola, and S. Atourian, "Improving Gil-Werman Algorithm for Running Min and Max Filters," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 5, pp. 526-529, May 1997.

[19] P. Soille, "Morphological Image Analysis Applied to Crop Field Mapping," *Image and Vision Computing,* vol. 18, no. 13, pp. 1025-1032, Oct. 2000, see electronic appendix at http://ams.egeo.sai.jrc. it/soille/ivc2000.

[20] J. Sanchiz, F. Pla, J.A. Marchant, and R. Brivot, "Structure from Motion Techniques Applied to Crop Field Imaging," *Image and Vision Computing,* vol. 14, pp. 353-363, 1996.

[21] T. Hague, J.A. Marchant, and N. Tillett, "Autonomous Robot Navigation for Precision Horticulture," *Proc. IEEE Int'l Conf. Robotics and Automation,* pp. 1880-1885 Apr. 1997.

[22] J.A. Marchant, R. Tillett, and R. Brivot, "Real-Time Segmentation of Plants and Weeds," *Real-Time Imaging,* vol. 4, pp. 243-253, 1998.

[23] G. Garibotto and L. Lambarelli, "Fast On-Line Implementation of Two Dimensional Median Filtering," *Electronic Letters,* vol. 15, no. 1, pp. 24-25, Jan. 1979.

[24] T. Huang, G. Yang, and G. Tang, "A Fast Two-Dimensional Median Filtering Algorithm," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 27, no. 1, pp. 13-18, Feb. 1979.

[25] P. Soille, "Grey Scale Convex Hulls: Definition, Implementation, and Application," *Mathematical Morphology and Its Applications to Image and Signal Processing,* H. Heijmans and J. Roerdink, eds., pp. 83-90, Kluwer Academic, 1998.

[26] P. Soille, "From Binary to Grey Scale Convex Hulls," *Fundamenta Informaticae,* vol. 41, nos. 1-2, pp. 131-146, Jan. 2000.

[27] P. Soille, *Morphological Image Analysis.* Berlin, Heidelberg, New York: Springer-Verlag, 1999.

[28] R. Jones and P. Soille, "Periodic Lines: Definition, Cascades, and Application to Granulometries," *Pattern Recognition Letters,* vol. 17, no. 10, pp. 1057-1063, Sept. 1996.

[29] H. Minkowski, "Volumen und Oberfläche," *Mathematische Annalen,* vol. 57, pp. 447-495, 1903.

[30] G. Matheron, *Random Sets and Integral Geometry.* New York: Wiley, 1975.

[31] B. Chaudhuri, "An Efficient algorithm for Running Window Pel Grey Level Ranking in 2D Images," *Pattern Recognition Letters,* vol. 11, no. 2, pp. 77-80, 1990.

[32] M. Van Droogenbroeck and H. Talbot, "Fast Computation of Morphological Operations with Arbitrary Structuring Elements," *Pattern Recognition Letters,* vol. 17, no. 14, pp. 1451-1460, 1996.
[33] G. Heygster, "Rank Filters in Digital Image Processing," *Computer Graphics and Image Processing,* vol. 19, no. 2, pp. 148-164, June 1982.
[34] J. Serra, "Introduction to Morphological Filters," *Image Analysis and Mathematical Morphology. Vol 2: Theoretical Advances,* J. Serra, ed., ch. 5, pp. 101-114. Academic Press, 1988.
[35] J. Serra and L. Vincent, "An Overview of Morphological Filtering," *Circuits Systems Signal Process,* vol. 11, no. 1, pp. 47-108, 1992.
[36] C. Ronse, "Erosion of Narrow Image Features by Combination of Local Low Rank and Max Filters," *Proc. Second IEE Int'l Conf. Image Processing and Its Applications,* pp. 77-81, 1986.
[37] C. Ronse and H. Heijmans, "The Algebraic Basis of Mathematical Morphology: II. Openings and Closings," *Computer Vision, Graphics, and Image Processing: Image Understanding,* vol. 54, no. 1, pp. 74-97, 1991.
[38] H. Heijmans, "Morphological Image Operators," *Advances in Electronics and Electron Physics,* Boston: Academic Press, 1994.
[39] S. Wilson, "Vector Morphology and Iconic Neural Networks," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 19, no. 6, pp. 1636-1644, 1989.
[40] P. Soille, "On Morphological Operators Based on Rank Filters," *Pattern Recognition,* pending publication.
[41] P. Whelan and P. Soille, "Watermark Extraction in Paper Samples," *Proc. Optical Eng. Soc. Ireland and Irish Machine Vision and Image Processing Joint Conf.,* D. Vernon, ed., pp. 287-299, Sept. 1998.
[42] A. Drimbarean, P. Soille, and P. Whelan, "Online Registration of Watermarks from Continuous Web Paper," *Proc. Irish Machine Vision and Image Processing Conf.,* P. Whelan, ed., pp. 267-279, 1999.
[43] E. Isaac and R. Singleton, "Sorting by Address Calculation," *J. ACM,* vol. 3, pp. 169-174, 1956.
[44] D. Saupe, "Algorithms for Random Fractals," *The Science of Fractal Images,* H. Peitgen and D. Saupe, eds., ch. 2, New York: Springer-Verlag, 1988.
[45] J. Farey, "On a Curious Property of Vulgar Fractions," *Philosophical Magazine,* 1816, see http://cut-the-knot.com/blue/FareyHistory.html for a historical note and a copy of Farey's original letter.
[46] G. Matheron, *Eléments pour une Théorie des Milieux Poreux.* Paris: Masson, 1967.
[47] J. Serra, *Image Analysis and Mathematical Morphology.* London: Academic Press, 1982.
[48] P. Maragos, "Pattern Spectrum and Multiscale Shape Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 701-716, July 1989.
[49] M. Werman and S. Peleg, "Min-Max Operators in Texture Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, no. 6, pp. 730-733, Nov. 1985.
[50] M. Donahue and S. Rokhlin, "On the Use of Level Curves in Image Analysis," *Computer Vision, Graphics, and Image Processing: Image Understanding,* vol. 57, no. 2, pp. 185-203, Mar. 1993.
[51] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, "Fingerprint Classification by Directional Image Partitioning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 5, pp. 402-421, May 1999.
[52] R. van den Boomgaard and R. van Balen, "Methods for Fast Morphological Image Transforms Using Bitmapped Binary Images," *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing,* vol. 54, no. 3, pp. 252-258, May 1992.
[53] R. Adams, "Radial Decomposition of Discs and Spheres," *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing,* vol. 55, no. 5, pp. 325-332, Sept. 1993.
[54] J.-F. Rivest, J. Serra, and P. Soille, "Dimensionality in Image Analysis," *J. Visual Comm. and Image Representation,* vol. 3, no. 2, pp. 137-146, 1992.
[55] P. Soille and J.-F. Rivest, "Dimensionality of Morphological Operators and Cluster Analysis," *Image Algebra and Morphological Image Processing IV,* E. Dougherty, P. Gader, and J. Serra, eds., pp. 43-53 July 1993.
[56] S. Sternberg, "Grayscale Morphology," *Computer Graphics and Image Processing,* vol. 35, pp. 333-355, 1986.
[57] P. Soille and R. Jones, "Periodic Lines: Fast Implementation and Extensions to Greyscale Structuring Elements and 3D Images," technical report, Fraunhofer IPK (Berlin)/CSIRO DMS (Sydney), 1995.
[58] P. Soille, "Morphological Operators with Discrete Line Segments," *Proc. Discrete Geometry for Computer Imagery 2000,* G. Borgefors, I. Nyström, and G. Sanniti di Baja, eds., pp. 78-98, 2000.
[59] K. Sivakumar, M. Patel, N. Ketharnavaz, Y. Balagurunathan, and E. Dougherty, "A Constant-Time Algorithm for Erosions/Dilations with Applications to Morphological Texture Feature Computation," *Real-Time Imaging,* vol. 6, pp. 223-239, 2000.
[60] R. Jones and P. Soille, "Periodic Lines and Their Applications to Granulometries," *Math. Morphology and Its Applications to Image and Signal Processing,* P. Maragos, W. Schafer, and M. Butt, eds., pp. 264-272, Kluwer Academic, 1996.

Pierre Soille received the engineering degree from the Université de Louvain in 1988. Research on the processing of digital elevation models and satellite images provided the material for the doctoral degree he received in 1992 at the same university and in collaboration with the Centre de Morphologie Mathématique of the Ecole des Mines de Paris. He then pursued research on image analysis and mathematical morphology at the CSIRO Mathematical and Information Sciences Division, Sydney, the Centre de Morphologie Mathématique, Fontainebleau, and the Abteilung Mustererkennung of the Fraunhofer-Institut IPK, Berlin. In 1994, he co-organized with Jean Serra the Second International Symposium on Mathematical Morphology. From 1995 to 1998, he was a lecturer and research scientist at the Ecole des Mines d'Alès and EERIE, Nîmes. In 1999, he was a senior research scientist at the BBSRC Silsoe Research Institute, United Kingdom, leading a project on the application of novel image analysis techniques to agri-food industries. Since December 1999, Dr. Soille has been with the EC Joint Research Centre, Space Applications Institute, Italy. His contributions to the analysis of images through mathematical morphology formed the basis of the "Habilitation à Diriger des Recherches" degree he received in 1997 from the Université Montpellier II. He has worked on many applied projects and contributed to research publications on image segmentation, interpolation, pattern recognition, shape analysis, efficient algorithms, and scientific/industrial applications. He has taught tutorials on mathematical morphology at the 11th and 14th International Conferences on Pattern Recognition as well as during several other international conferences. He is author of the book *Morphological Image Analysis* recently published in English and German.

Hugues Talbot received the engineering degree from Ecole Centrale de Paris in 1989, the Diplôme d'Etudes Approfondies from University Paris VI in 1990 and the PhD degree from the Centre de Morphologie Mathématique, Ecole des Mines de Paris, in 1993, in collaboration with the Massachusetts Institute of Technology, Cambridge, and the Isover Saint Gobain corporation, under the guidance of Jean Serra and Dominique Jeulin. He has been affiliated with CSIRO, Mathematical and Information Sciences, Sydney, since 1994. He has worked on many applied projects and has contributed to more than 20 publications in international journals and conferences. His research interests include image segmentation, linear structure analysis, symmetry, and texture analysis and algorithms.

Appendix E

Path Openings and Closings

This article was published in JMIV as Heijmans et al. (2005).

```
@article{HeijmansEtal2005,
Author = {Heijmans, H. and Buckley, M. and Talbot, H.},
Journal = {Journal of Mathematical Imaging and Vision},
Pages = {107-119},
Title = {Path openings and closings},
Volume = {22},
Year = {2005}}
```

Journal of Mathematical Imaging and Vision 22: 107–119, 2005

Path Openings and Closings

HENK HEIJMANS*
CWI, Amsterdam, The Netherlands
henk.heijmans@cwi.nl

MICHAEL BUCKLEY AND HUGUES TALBOT†
CSIRO Mathematical and Information Science, Locked Bag 17, North Ryde, NSW 1670 Australia
michael.buckley@csiro.au
hugues.talbot@csiro.au

Abstract. This paper lays the theoretical foundations to path openings and closings.

The traditional morphological filter used for the analysis of linear structures in images is the union of openings (or the intersection of closings) by linear segments. However structures in images are rarely strictly straight, and as a result a more flexible approach is needed.

An extension to the idea of using straight line segments as structuring elements is to use constrained paths, i.e. discrete, one-pixel thick successions of pixels oriented in a particular direction, but in general forming curved lines rather than perfectly straight lines. However the number of such paths is prohibitive and the resulting algorithm by simple composition is inefficient.

In this paper we propose a way to compute openings and closings over large numbers of constrained, oriented paths in an efficient manner, suitable for building filters with applications to the analysis of oriented features, such as for example texture.

Keywords: oriented features, algebraic morphological filters, flexible linear morphological filters

1. Introduction

Practitioners of mathematical morphology are familiar with the importance of the structuring element in morphological and algebraic openings and closings. In spite of the infinite variety of available structuring elements, very few kinds of structuring elements are used in practice outside of a few specialized applications. The unit ball structuring elements of the discrete grid (e.g.: diamond, square and hexagon) define a first family of common structuring elements, useful for basic filtering, granulometries, etc.

Probably the second most used structuring element family is generated by some instance of the discrete line segment, which is used when linear and oriented structures are present in an application [8]. This limited choice can be at least partly blamed on the dearth of truly efficient algorithms for more arbitrary structuring elements [9].

However most structures in real-world images are not perfectly straight, and therefore using line segments as structuring elements in openings and closings can be inadequate in the common situation where there exist narrow, locally oriented features in an image of interest. In this case one might be interested in using structuring elements that are themselves narrow and oriented, but not perfectly straight. Unfortunately generating useful morphological filters in the usual way

*This paper was written at CSIRO MIS while on leave from CWI.
†Author to whom all correspondence should be addressed.

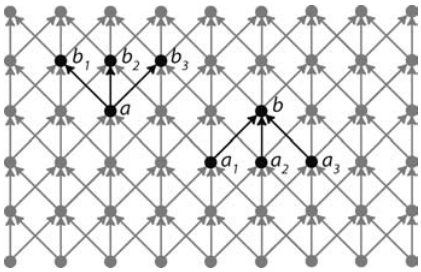


Figure 1. b_1, b_2, b_3 are successors of a and a_1, a_2, a_3 are the predecessors of b.

by composition leads to computationally expensive and impractical algorithms.

An earlier method for path-based filtering using morphological ideas was devised by Vincent [10] and provided the inspiration for the early work on path openings and closings. However Vincent's filter is based on local *sums* of pixels values whereas the path openings and closings described in this paper are based on local *minima* and *maxima*. Vincent's filter therefore behaves more like an oriented smoothing, and does not constitute a morphological filter.

In this paper we introduce the concept of path openings and closings, i.e. morphological filters that use families of structuring elements consisting of variously constrained paths, for which there exists algorithms as efficient as those using the usual families of straight line segments.

Path openings were originally proposed in [1] in an algorithmic, practical but theoretically incomplete manner. Here we are more concerned with laying down the theoretical foundations of these useful filters.

2. Adjacencies, Dilations and Paths

2.1. Adjacencies

Let E be a given set of points representing pixel locations. Define a *directed graph* on these points via a binary adjacency relation '$\mapsto$'. Specifically, $x \mapsto y$ means that that there is an edge going from x to y. If $x \mapsto y$, we call y a *successor* of x and x a *predecessor* of y. These concepts are illustrated in Fig. 1. Here b_1, b_2, b_3 are successors of a and a_1, a_2, a_3 are the predecessors of b.

The relation '$\mapsto$' is, in general, neither reflexive nor symmetric.[1] We show some examples in Fig. 2.

Note that in the first three examples in Fig. 2, the adjacency relation is periodic away from the borders. The fact that we can choose any adjacency as a starting point enables us to handle the border in a consistent and flexible manner. Note also the following major difference between the adjacency relation in Fig. 2(a) and (c) and the one in (b). In (a) and (c), the graph structure is translation invariant with respect to any translation (again away from the borders), whereas in (b), this is only true if translation takes place over an even number of rows. Rephrased in terms of the dilation (see next subsection), this means that the structuring element is different at odd and at even rows. We will briefly address the issue of choosing the adjacency relation in Section 9.

2.2. Dilations

Using the adjacency relation we have, for each point x, a set of its successors with respect to '$\mapsto$'. Denote this by $\delta(\{x\})$. That is,

$$\delta(\{x\}) = \{y \in E : x \mapsto y\}.$$

This can be generalised to arbitrary subsets X of E as follows:

$$\delta(X) = \{y \in E : x \mapsto y \text{ for some } x \in X\}.$$

In other words, $\delta(X)$ comprises all points which have a predecessor in X. In similar fashion we define the set $\check{\delta}(X)$ as the set of points which have a *successor* in X.

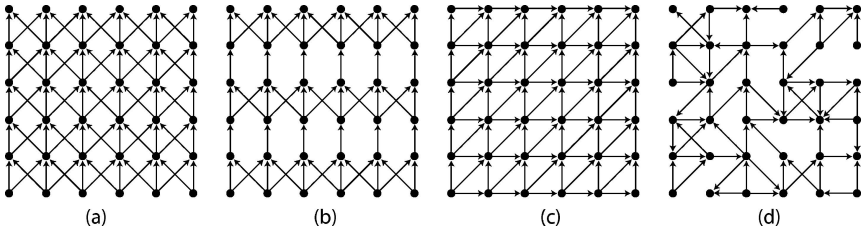


Figure 2. Directed graphs.

The operators δ and $\check{\delta}$ have the property that $\delta(\cup_i X_i) = \cup_i \delta(X_i)$ for any sets X_i. This is the defining algebraic property of a *dilation*. Hence we refer to these operators as dilations.

Here we have first provided the adjacency relation '$\mapsto$' and defined the dilation in terms of '$\mapsto$'. It is straightforward to see that we might as well start with any dilation δ on $\mathcal{P}(E)$ and define the adjacency relation as

$$x \mapsto y \quad \text{if } y \in \delta(\{x\}).$$

Both approaches are equivalent and it is a matter of taste which one is taken.

2.3. Paths

The L-tuple $\boldsymbol{a} = (a_1, a_2, \ldots, a_L)$ is called a *path of length* L if $a_k \mapsto a_{k+1}$, or equivalently, if

$$a_{k+1} \in \delta(\{a_k\}), \quad \text{for } k = 1, 2, \ldots, L-1.$$

Henceforth we refer to such a path as a δ-path of length L. It is evident that $\boldsymbol{a} = (a_1, a_2, \ldots, a_L)$ is δ-path if and only if the reverse path $\check{\boldsymbol{a}} = (a_L, a_{L-1}, \ldots, a_1)$ is a $\check{\delta}$-path and obviously, both paths have the same length L. We denote the set of all δ-paths of length L by Π_L and the set of all $\check{\delta}$-paths of length L by $\check{\Pi}_L$. Given a path $\boldsymbol{a}$ in E, we denote by $\sigma(\boldsymbol{a})$ the set of its elements:

$$\sigma(a_1, a_2, \ldots, a_L) = \{a_1, a_2, \ldots, a_L\}.$$

The set of δ-paths of length L contained in a subset X of E is denoted by $\Pi_L(X)$, i.e.,

$$\Pi_L(X) = \{a \in \Pi_L : \sigma(a) \subseteq X\}$$

and the $\check{\delta}$-paths of length L in X by $\check{\Pi}_L(X)$.

3. Path Opening

We define the set $a_L(X)$ as the union of all δ-paths of length L contained in X:

$$\alpha_L(X) = \bigcup \{\sigma(\boldsymbol{a}) : \boldsymbol{a} \in \Pi_L(X)\}.$$

It is not difficult to establish that the operator α_L has the algebraic properties of an *opening*, specifically increasingness, anti-extensivity and idempotence, and we call it the *path opening*.

Moreover in the cases of the usual periodic adjacencies, α_L is the supremum of morphological openings by a certain class of structuring elements, and the number of structuring elements in this class grows exponentially with L. For example, for an unbounded domain and the adjacency shown in Fig. 1(a), there are 3^{L-1} paths of length L beginning from any point. None of these is a translation of another, and hence as structuring elements for morphological openings or closings they are distinct. The path opening a_L can be shown to be the supremum of morphological openings by all of these 3^{L-1} structuring elements. In Section 4 we demonstrate how this opening may be computed with cost which grows linearly with L, even though the number of structuring elements implicitly involved grows exponentially with L. This is analogous to dynamic programming algorithms for shortest-paths in which the minimal path from an exponential collection of paths is computed in linear time.

We can define the reciprocal path opening $\check{\alpha}_L(X)$ in a similar way. Since $\boldsymbol{a} \in \Pi_L(X)$ iff $\check{\boldsymbol{a}} \in \check{\Pi}_L(X)$ and $\sigma(\boldsymbol{a}) = \sigma(\check{\boldsymbol{a}})$ we get immediately that

$$\alpha_L = \check{\alpha}_L.$$

It is obvious that $\alpha_1 = \mathsf{id}$. We can show that that

$$\alpha_{L+1} \leq \alpha_L \quad \text{for} \quad L \geq 1.$$

To prove this, assume that $x \in \alpha_{L+1}(X)$. Thus there is a δ-path $(a_1, a_2, \ldots, \alpha_{L+1})$ of length $L+1$ which contains x and lies inside X. But then both δ-paths $(a_1, a_2, \ldots, a_L)$ and $(a_2, a_3, \ldots, a_{L+1})$ of length L lie inside X and at least one of them must contain the point x. This proves that $x \in \alpha_L(X)$, too. In Fig. 3 we show an example of a path opening where $L = 6$.

4. Computation of the Path Opening

In this section we define "first-point sets", $\psi_k(X)$ and derive relationships between these and the opened sets $\alpha_L(X)$ which allow feasible computation of path openings.

4.1. Path Decomposition

By definition, $x \in \alpha_L(X)$ iff there exists a δ-path $\boldsymbol{a} \in \Pi_L(X)$ that contains x, i.e., $x = a_k$ for some k between

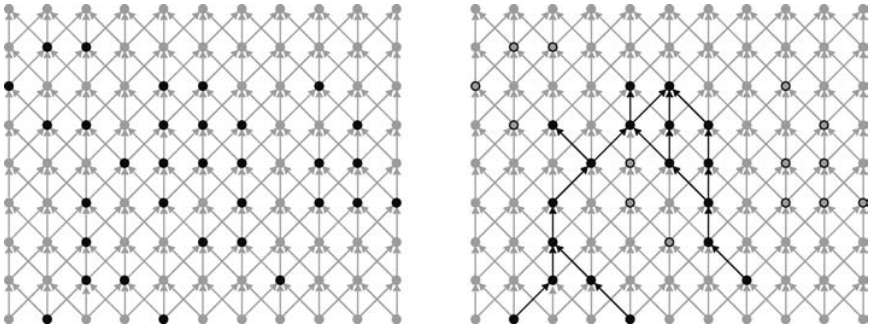


Figure 3. A set $X \subseteq E$ (black points at the left) and its opening $a_6(X)$ (black points at the right). Unfilled points at the right have been discarded. Adjacency graph is in light grey and identical to that of Fig. 2a.

1 and L. In that case we have

$$\begin{aligned}(a_1, a_2, \ldots, a_{k-1}, x) \in \Pi_k(X) \quad \text{and} \\ (x, a_{k+1}, \ldots, a_L) \in \Pi_{L-k+1}(X).\end{aligned} \tag{1}$$

The first condition can be rewritten as

$$(x, a_{k-1}, a_{k-2}, \ldots, a_1) \in \check{\Pi}_k(X). \tag{2}$$

We define the operator ψ_k as

$$\psi_k(X) = \{a_1 : \boldsymbol{a} \in \Pi_k(X)\},$$

that is, $\psi_k(X)$ contains the first point of every δ-path of length k in X. The operator $\check{\psi}_k$ is defined analogously. Obviously, $\psi_1 = \check{\psi}_1 = \mathsf{id}$.

Now the first condition in (1), which is equivalent to (2), can be written as $x \in \check{\psi}_k(X)$, and the second condition can be written as $x \in \psi_{L-k+1}(X)$. Combined, they give

$$x \in \check{\psi}_k(X) \cap \psi_{L-k+1}(X) \text{ for some } k.$$

By taking the supremum over all the k, this is equivalent to

$$\alpha_L(X) \subseteq \bigcup_{k=1}^{L} (\check{\psi}_k(X) \cap \psi_{L-k+1}(X))$$

By definition of $\check{\psi}_k$ and ψ_k, the union of all of these contains the union of all the paths of length L contained in X. By definition of α_L we have:

$$\bigcup_{k=1}^{L} (\check{\psi}_k(X) \cap \psi_{L-k+1}(X)) \subseteq \alpha_L(X).$$

Using the complete lattice notation, we have shown that

$$a_L = \bigvee_{k=1}^{L} (\check{\psi}_k \wedge \psi_{L-k+1}). \tag{3}$$

Note that $a_2 = \mathsf{id} \wedge (\delta \vee \check{\delta})$, which is known in the literature as the *annular opening* [2,6].

Finally we note that the semi-group property

$$\psi_k \psi_l = \psi_{k+l-1}$$

holds for all $k, l \geq 1$. The proof of this is straightforward.

4.2. *Recursive Structure of the Operators ψ_k*

We will prove below that the following relations hold:

$$\psi_{k+1} = \mathrm{id} \wedge \check{\delta}\psi_k \quad \text{and} \quad \check{\psi}_{k+1} = \mathrm{id} \wedge \delta\check{\psi}_k\,. \tag{4}$$

The decomposition of a_L in (3) together with the iterative formulas in (4) provide an efficient algorithm for the computation of the path opening a_L. As shown in Algorithm 1, we compute the path opening $\alpha_L(X)$ given the set X and dilation operators δ and $\check{\delta}$. If we have a finite domain E of size N then unions and intersections can be computed in $\mathrm{O}(N)$ operations. For simple periodic adjacencies such as those in Fig. 2(a)–(c), the dilations δ and $\check{\delta}$ can also be computed in $\mathrm{O}(N)$ operations. This gives a total cost for Algorithm 1 of $O(LN)$ for both operations and memory.

Algorithm 1 (Binary Path-Opening).

// Recursive computation of $Y_k = \psi_k(X)$ and
// $\breve{Y}_k = \breve{\psi}_k(X)$ using Eq. (4)
$Y_1 = \breve{Y}_1 = X$
for $k = 1$ **to** $L - 1$
$\quad Y_{k+1} = X \cap \breve{\delta}(Y_k)$
$\quad \breve{Y}_{k+1} = X \cap \delta(\breve{Y}_k)$
end

// Construction of opening $A_L = \alpha_L(X)$ using Eq. (3)
$A_L = \emptyset$
for $k = 1$ **to** L
$\quad A_L = A_L \cup (Y_k \cap \breve{Y}_{L-1+k})$
end

We will prove only the first identity in (4) as the second is nothing but its reciprocal version. To prove '$\leq$' assume that $x \in \psi_{k+1}(X)$. This means that there exist $a_2, \ldots, a_{k+1}$ such that $(x, a_2, \ldots, a_{k+1}) \in \Pi_{k+1}(X)$. Now $(a_2, \ldots, a_{k+1}) \in \Pi_k(X)$ and $x \in \breve{\delta}(\{a_2\})$. Since $a_2 \in \psi_k(X)$ this yields that $x \in \breve{\delta}(\psi_k(X))$, and we conclude that $x \in X \cap \breve{\delta}(\psi_k(X))$.

To prove '$\geq$', let $x \in (\mathsf{id} \wedge \breve{\delta}\psi_k)(X)$, i.e., $x \in X$ and $x \in \breve{\delta}(\{y\})$ with $y \in \psi_k(X)$. The latter means that there exist $a_2, \ldots, a_k$ such that $(y, a_2, \ldots, a_k) \in \Pi_k(X)$. Now $(x, y, a_2, \ldots, a_k) \in \Pi_{k+1}(X)$, which yields that $x \in \psi_{k+1}(X)$.

The path opening α_L depends strongly upon the dilation, or equivalently, the adjacency relation. This is clearly seen in Fig. 4 where we have computed the opening $\alpha_5(X)$ of a set X for three different adjacencies. A union of openings is an opening [5], therefore we can, for example, take the union of the first two openings in Fig. 4, i.e., two figures in the middle, to get an opening that allows both horizontal and vertical oriented paths. Note however, that this is not the same as combining both adjacencies into one and computing the opening with respect to this new adjacency.

5. Opening Transform

Often, we are interested in all openings $\alpha_L(X)$ of a set X for a range of values of L rather than for a single value only. For example, it is quite common that we do not know beforehand which L to choose in a particular application. In such cases it may be more efficient to compute the so-called *opening transform* of the image. Given a set $X \subseteq E$ and an ordered family of openings $\mathcal{A} = \{\alpha_L\}$, the opening transform A_X of X with respect to $\mathcal{A}$ is a function mapping the domain E into $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$ such that its threshold sets correspond with the various openings $\alpha_L(X)$: see (6) below for an exact formulation.

Throughout this section we assume that there exist only finite acyclic paths. More precisely we assume that there exists an integer $N \geq 1$ such that $\delta^N(\{x\}) = \emptyset$, for every $x \in E$, i.e., $\delta^N(E) = \emptyset$. Note, however, that this assumption does not necessarily mean that E is finite. Furthermore, we take N to be the smallest integer with this property. Thus the maximal length of a path in E is N. Define $\lambda(x)$ as the maximal length of a δ-path with begin-point x:

$$\begin{aligned}\lambda(x) &= \max\{L \geq 1 : \exists\, a \in \Pi_L \text{ such that } a_1 = x\} \\ &= \max\{L \geq 1 : x \in \breve{\delta}^{L-1}(E)\}.\end{aligned}$$

Obviously, if $x \mapsto y$ then $\lambda(x) \geq \lambda(y) + 1$. Moreover, it is not difficult to prove that

$$\lambda(x) = 1 + \max\{\lambda(y) : x \mapsto y\},$$

where the maximum is taken to be zero if x has no successors. Similarly $\breve{\lambda}(x)$ is the maximal length of a path with endpoint x. Then

$$\Lambda(x) = \lambda(x) + \breve{\lambda}(x) - 1, \tag{5}$$

is the length of the longest path that contains x. This is a direct consequence of Eqs. (1) and (2).

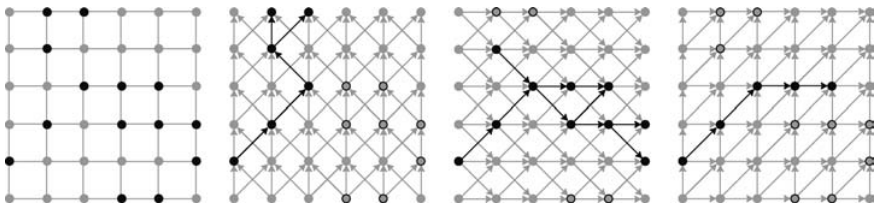


Figure 4. A set $X \subseteq E$ (left) and its opening $\alpha_5(X)$ for three different adjacencies.

Define the disjoint partition $E_1, E_2, \ldots, E_N$ of E by

$$E_L = \{x \in E : \lambda(x) = L\}.$$

Similarly $\check{E}_L$ is defined with $\check{\lambda}$ instead of λ. In the figures at the left in Fig. 5 we depict this partition for two different adjacency relations; here the arrows indicate the relation $x \mapsto y$. Note that E_L and $\check{E}_L$ can be computed easily via a distance transform algorithm.

Now we define a function $F_X : E \to \mathbb{Z}_+$ by means of Algorithm 2 which resembles a geodesic propagation algorithm.

Algorithm 2.

$F_X = 0$ on E // Initialisation
for $k = 1$ **to** N
  for $x \in E_k \cap X$
    $F_X(x) = 1 + \max\{F_X(y) \mid y \mapsto x\}$
  end
end

The second column of Fig. 5 shows the function F_X for a given image X (grey pixels). In a similar way we can define $\check{F}_X$ by using the partition $\check{E}_1, \check{E}_2, \ldots, \check{E}_N$.

The following lemma shows that $\psi_k(X)$ can be obtained by thresholding of F_X.

Lemma 5.1. *With the definitions given before we have*

$$\psi_k(X) = \{x \in E : F_X(x) \geq k\} \quad \textit{and}$$
$$\check{\psi}_k(X) = \{x \in E : \check{F}_X(x) \geq k\},$$

for $k = 1, 2, \ldots, N$.

Proof: Let $x \in E_k$ and suppose that $F_X(x) = l$. Obviously, $l \leq k$ and there must exist a δ-path $x = a_1, a_2, \ldots, a_l$ in $\Pi_l(X)$ such that $a_i \in E_{k-i+1}$ and $F_X(a_i) = l - i + 1$. This implies that $x \in \psi_l(X)$. Conversely, if $x \in \psi_l(X)$, then there is a path $x = a_1, a_2, \ldots, a_l$ in $\Pi_l(X)$. Now if $x \in E_k$, where $k \geq l$, then $a_i \in E_{k-i+1}$ and $F_X(a_i) = l - i + 1$. □

This lemma can be used to prove the following result.

Proposition 5.2. *The function* $A_X = F_X + \check{F}_X - 1$ *is the opening transform of* X, *that is*

$$\alpha_L(X) = \{x \in E : A_X(x) \geq L\}, \tag{6}$$

for every $L \geq 1$ *and* $X \subseteq E$.

Proof: We use the expression for α_L in (3) which says that $x \in \alpha_L(X)$ implies that $x \in \psi_k(X) \cap \check{\psi}_{L-k+1}(X)$ for some $k = 1, 2, \ldots, L$. Therefore, $F_X(x) \geq k$ and $\check{F}_X(x) \geq L - k + 1$, which yields that $A_X(x) \geq L$.

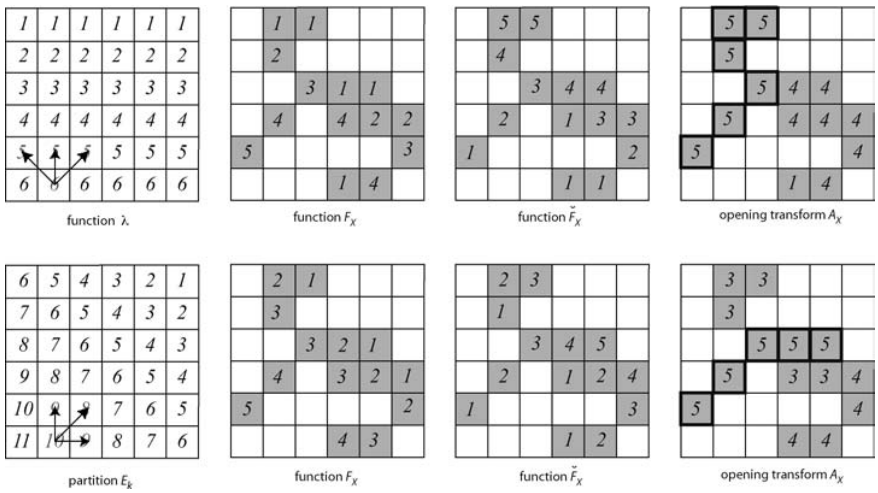


Figure 5. Computation of the opening transform for two different adjacencies indicated by the arrows in the first column, which also shows the partition of E. The second and third columns depict the functions F_X and $\check{F}_X$ for the set X represented by the grey pixels. The right column shows the opening transform A_X and the opened set $\alpha_5(X)$ represented by the pixels with the thick boundaries.

This proves '$\subseteq$' in (6). The converse is proved similarly. □

In Fig. 5, the algorithm for the opening transform is shown for two different adjacencies, namely (*a*) and (*c*) in Fig. 2.

6. Border Issues

Throughout this section, E will be a finite rectangular window within $\mathbb{Z}^2$, and the adjacency on E is the restriction of a periodic adjacency on $\mathbb{Z}^2$ like in Fig. 2 (a)–(c). The *inward boundary* of E, denoted by ∂E, is the set of points in E which have a predecessor outside E:

$$\partial E = \{x \in E : \exists\, y \in E^c \text{ such that } y \mapsto x\},$$

where E^c is the complement of E. The *outward boundary* of E, denoted by $\breve{\partial} E$, is given by

$$\breve{\partial} E = \{x \in E : \exists\, y \in E^c \text{ such that } x \mapsto y\},$$

There are various ways to deal with the border problem:

(a) We can simply ignore the existence of the borders and treat paths that contain boundary points in the same way as any other path. In fact, this is the choice that we have implicitly made so far.

(b) The other extreme is to set the length of a path that crosses the border to $+\infty$, meaning that all points on such a path are contained in every opening $\alpha_L(X)$. In fact, this choice means that we extend X outside E by adding all points in E^c.

(c) An intermediate option is to try to compensate for the points cut off by restricting to a finite window: we replace the computed length L of a path that has a begin-point in ∂E or an endpoint in $\breve{\partial} E$ by $h(L)$. One possible choice for h would be $h(L) = 2L$. Such a choice could be justified by the presumption that on average only half of the path falls inside the window. Another possibility is to add a fixed compensation to the length of a border-crossing path, i.e., $h(L) = L + L_0$. Note, that one might use different compensation functions for paths that start on ∂E and end on $\breve{\partial} E$.

(d) A possibility which is easy to implement is to enlarge the window E with a border B of thickness L_0. Denote by α'_L the corresponding path opening on $\mathcal{P}(E')$, where $E' = E \cup B$ denotes the enlarged window. Thus we can compute α'_L according to the algorithm given in the previous sections. Define the opening α_L on $\mathcal{P}(E)$ by

$$\alpha_L(X) = \alpha'_L(X \cup B) \cap E.$$

In Fig. 6 we show that in this case, two paths which were originally disjoint may considered to be part of the same path which lies partially outside the window.

7. Incomplete Path Openings

7.1. Motivation

In practical applications, to increase the discriminatory power of path openings one might need to increase L. However as L increases so does the probability of any path containing noise. To increase the robustness and flexibility of path openings, it is useful to allow a limited number of noise pixels to be ignored. This is the basic idea behind rank-max openings [4], which have been proven to work well in filtering applications [3], in particular with line structuring elements [8].

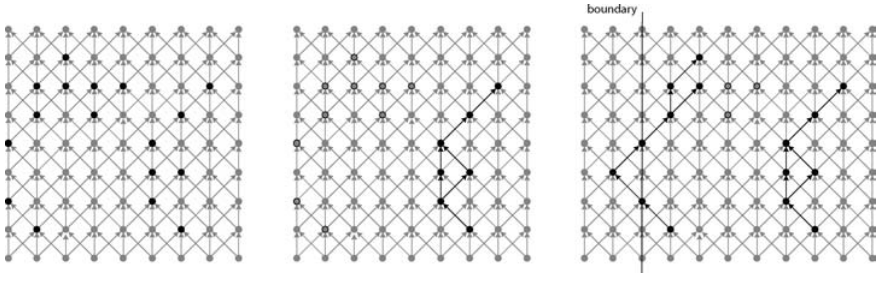


Figure 6. Opening $\alpha_6(X)$ of set X (left) according to option (*a*) (middle) and (*d*) (right).

7.2. *Framework*

The path opening $\alpha_L(X)$ of a set X comprises the union of all length-L paths contained inside X. We can relax this condition by allowing up to k vertices to be outside X. That is, we demand that at least $L-k$ out of L vertices of the path must lie inside X. This yields a so-called *incomplete path opening* $\alpha_L^k(X)$. We present a formal definition below.

Define $\Pi_L^k(X)$ as the collection of length-L paths in E which contain at most k points outside X:

$$\Pi_L^k(X) = \{a \in \Pi_L : |\sigma(a) \cap X^c| \leq k\}.$$

Note that this definition only makes sense for $0 \leq k \leq L$, and that

$$\begin{aligned}\Pi_L(X) = \Pi_L^0(X) \subseteq \Pi_L^1(X) \subseteq \cdots \subseteq \Pi_L^{L-1}(X)\\ \subseteq \Pi_L^L(X) = \Pi_L.\end{aligned}$$

For $0 \leq k \leq L-1$ define the *incomplete path opening*

$$\alpha_L^k(X) = \bigcup \big\{\sigma(a) \cap X : a \in \Pi_L^k(X)\big\}.$$

It is obvious that

$$\alpha_L^0 \leq \alpha_L^1 \leq \cdots \leq \alpha_L^{L-1},$$

that $\alpha_L^0 = \alpha_L$ and

$$\alpha_L^{L-1}(X) = \{x \in X : \Lambda(x) \geq L\},$$

where $\Lambda(x)$ was defined in (5). Putting

$$\bar{E}_L = \{x \in E : \Lambda(x) \geq L\},$$

we get that

$$\alpha_L^{L-1}(X) = X \cap \bar{E}_L.$$

Furthermore, for $0 \leq k \leq L$ we define

$$\psi_L^k(X) = \big\{a_1 : a \in \Pi_L^k(X)\big\}.$$

We have $\psi_L^0 = \psi_L$ and

$$\psi_L^L(X) = \{x \in E : \check{\lambda}(x) \geq L\} = \bigcup_{k \geq L} \check{E}_k.$$

Note that ψ_L^k is defined for $0 \leq k \leq L$ while α_L^k is defined only for $0 \leq k \leq L-1$.

We will now express ψ_{L+1}^{k+1} in terms of ψ_L^{k+1} and ψ_L^k. Observe that $x \in \psi_{L+1}^{k+1}(X)$ if there exists $a = (a_1, \ldots, a_L)$ such that $(x, a_1, \ldots, a_L) \in \Pi_{L+1}$ and either $x \in X$ and $a \in \psi_L^{k+1}(X)$ or $a \in \psi_L^k(X)$. We have shown that

$$\psi_{L+1}^{k+1} = \big(\mathsf{id} \wedge \check{\delta}\psi_L^{k+1}\big) \vee \check{\delta}\psi_L^k. \tag{7}$$

This equation holds for $k = 0, 1, \ldots, L-1$. If we take ψ_L^{-1} to be the empty set, then the equation holds also for $k = -1$, and in fact reduces to (4).

We now derive a more general form of (3). Consider a point $x \in \alpha_L^k(X)$, where $0 \leq k \leq L-1$. Thus $x \in X$ and there is a path $a \in \Pi_L$ with $x \in \sigma(a)$ such that $|\sigma(a) \cap X^c| \leq k$. Assume that $a_l = x$. Now a is the concatenation of the sequences $\boldsymbol{b} = (a_1, \ldots, a_{l-1}, x)$ and $\boldsymbol{c} = (x, a_{l+1}, \ldots, a_L)$. Define $j = |\sigma(\boldsymbol{b}) \cap X^c|$ which implies $|\sigma(\boldsymbol{c}) \cap X^c| \leq k - j$. We conclude that

$$x \in \check{\psi}_l^j(X) \cap \psi_{L-l+1}^{k-j}(X).$$

Since the length of $\boldsymbol{b}$ is l and $x \in X$ we have $0 \leq j \leq l-1$. Similarly since the length of $\boldsymbol{c}$ is $L+1-l$ we have $0 \leq k-j \leq L-l$. Together these imply that $0 \leq j \leq k$ and

$$j+1 \leq l \leq L+j-k$$

and we conclude that

$$\alpha_L^k = \mathsf{id} \wedge \bigvee_{j=0}^{k} \bigvee_{l=j+1}^{L+j-k} \big(\check{\psi}_l^j \wedge \psi_{L-l+1}^{k-j}\big) \tag{8}$$

for $0 \leq k \leq L-1$. Observe that this expression reduces to the one in (3) if $k = 0$.

The following is an algorithm for computation of the incomplete path opening. The computational cost for this algorithm is $O(kLN)$, with N the number of elements in space E.

Algorithm 3 (Incomplete Binary Path-Opening).

// Initialisation of bottom rows ($j = 0$) and diagonals ($j = l$)
// of arrays $Y_l^j = \psi_l^j(X)$ and $\check{Y}_l^j = \check{\psi}_l^j(X)$.
// Remainder of arrays Y_l^j and $\check{Y}_l^j$.
// Using Eq. (7).
for $j = 0$ **to** $k-1$

for $l = j$ **to** $L + j - k - 1$
$Y_{l+1}^{j+1} = \left(X \cap \check{\delta} Y_l^{j+1}\right) \cup \check{\delta} Y_l^j$
$\check{Y}_{l+1}^{j+1} = \left(X \cap \delta \check{Y}_l^{j+1}\right) \cup \delta \check{Y}_l^j$
end
end
// Construction of incomplete opening $A_L^k = \alpha_L^k(X)$
// Using Eq. (8).
$A_L^k = \emptyset$
for $j = 0$ **to** k
for $l = j + 1$ **to** $L + j - k$
$A_L^k = A_L^k \cup \left(Y_l^j \cap \check{Y}_{L-l+1}^{k-j}\right)$
end
end
$A_L^k = A_L^k \cap X$

7.3. *Illustration*

Figure 7 is a real example of the use of incomplete path openings in the binary case. In this image we have a thin glass fibre observed under an electron microscope with some noise present. A normal (i.e. complete) path opening deletes most of the noise together with the middle part of the fibre. An incomplete path opening with a small tolerance still deletes most of the noise while keeping the middle part of the fibre untouched.

The result of an incomplete path closing in the grey-level case is shown on Fig. 8(e) and discussed in the final section.

8. The Grey-Scale Case

In this section we extend the results developed in the previous sections to the grey-scale case, with the exception of the opening transform which is not defined in the grey-scale case to the best of our knowledge.

First we define the grey-scale analogue, $\Pi_L^t(I)$, of the path collection $\Pi_L(X)$. This involves an additional parameter t representing the grey-level of the path:

$$\Pi_L^t(I) = \{a \in \Pi_L : I(a_k) \geq t,\ k = 1, 2, \ldots, L\}$$

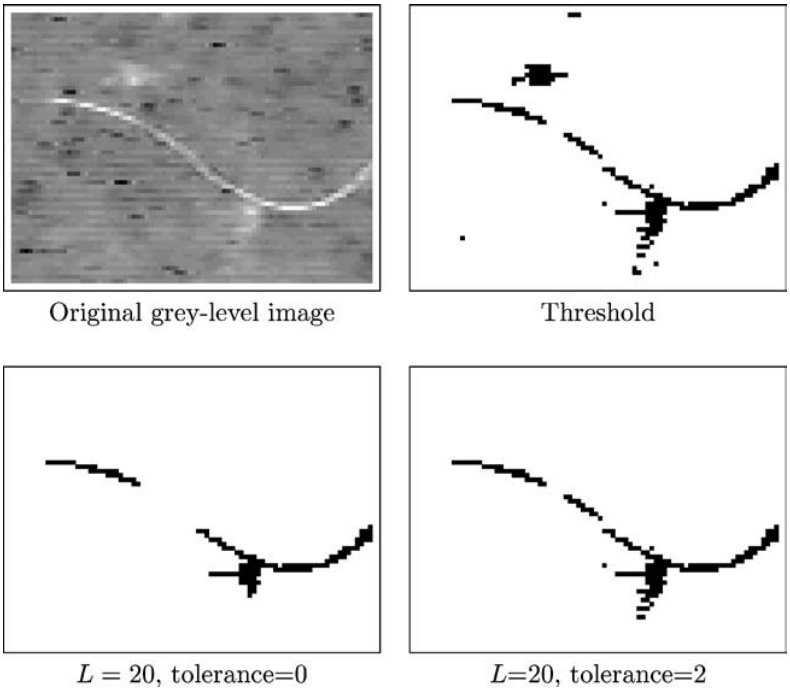
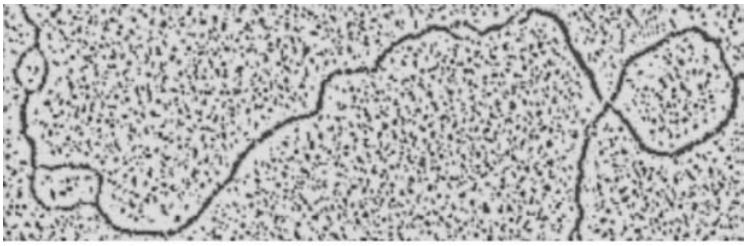
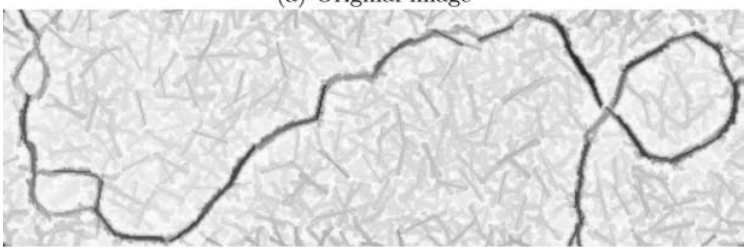


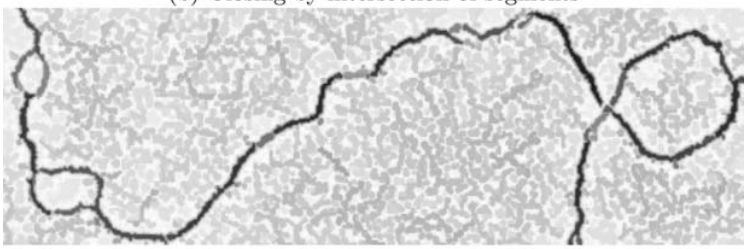
Figure 7. Glass fibre electron micrograph (a) and a threshold (b). Comparison between complete (c) and incomplete (d) path opening. With the incomplete path opening more of the fibre is retained.



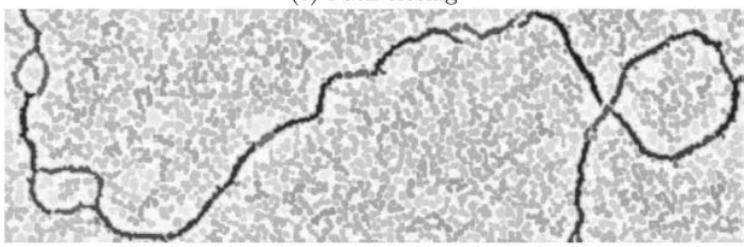
(a) Original image



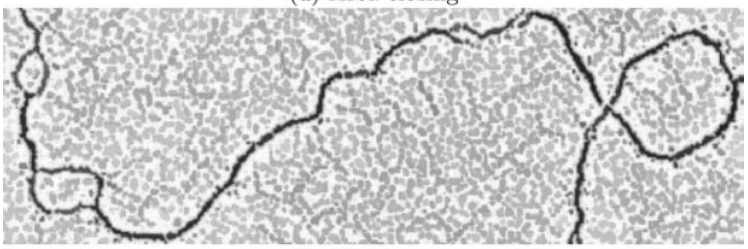
(b) Closing by intersection of segments



(c) Path closing



(d) Area closing



(e) Incomplete path closing

Figure 8. Example of path closing compared with closing with segments and area closing.

where

It is easy to see that

$$\Pi_L^t(I) = \Pi_L(X_t(I))$$

where

$$X_t(I) = \{x \in E : I(X) \geq t\}$$

is the *level set* of I at value t. Let Ψ_k be the flat extension of ψ_k defined by means of the level set construction, and let $\mathcal{A}_L$ be the flat extension of α_L.

For a domain E and range T define $\mathrm{Fun}(E, T) = T^E$ to be the space of grey-level functions from E to T.

Proposition 8.1. *The operator Ψ_k and the opening $\mathcal{A}_L$ on Fun(E, T) are, respectively, given by*

$$\Psi_k(I)(x) = \max\{t \in T : (x, a_2, \ldots, a_k) \in \Pi_k^t(I) \text{ for some } a_2, \ldots, a_k \in E\}$$

$$\mathcal{A}_L(I)(x) = \max\{t \in T : x \in \sigma(\boldsymbol{a}) \text{ for some } \boldsymbol{a} \in \Pi_L^t(I)\}$$

From the theory on flat function operators [2] we know that the expressions in (3) and (4) carry over immediately to the function case. Therefore the grey-scale opening $\mathcal{A}_L(I)$ of an image I may be computed by an algorithm which is a straightforward generalisation of Algorithm 1. Specifically, in Algorithm 1 we replace X by I, $\emptyset$ by $-\infty$, $\cup$ by $\vee$, $\cap$ by $\wedge$ and the dilations δ and $\check{\delta}$ by their flat extensions.

9. Example

Figure 8 is an example of path closing compared with other methods. We chose to illustrate with a closing rather than an opening because of the better contrast in the printing process, but the same conclusions would apply to both. The parameter choices for each operation cannot be made strictly identical, we chose them so that the outputs would be comparable.

Figure 8(a) is the grey-level original 500×160 image. This is an image of DNA (the long thin structure) observed in a scanning electron microscope. The objective is to separate the DNA from the noisy background, and we use various closings as pre-processing filters.

Figure 8(b) is the result of applying a closing by intersection of 44 segments of length 23 pixels, each in a different direction, approximately uniformly oriented (subject to the digital grid). As can be seen, after the application of this closing the background is mostly suppressed, but so is some of the DNA.

Figure 8(c) is a path closing with path length of 33 pixels, the specific form of which is given below. While this is longer than that of the straight segments family in (b), the shape and contrast of the DNA is well preserved, while the background is substantially suppressed.

Figure 8(d) is an area closing with parameter 50 square pixels. The DNA is well preserved as in (c) but the background is not as effectively filtered out.

Figure 8(e) is an incomplete path-closing with path length 33 and tolerance 2. This allows more of the DNA to be preserved, however the result is that some of the noise close to the DNA is not filtered out.

As can be seen, the path closing (c) was able to better preserve the shape of the object of interest than the closing with segments, while removing more of the unwanted background than the area closing, which was the intended behaviour.

To produce the closing shown in Fig. 8(c), four closings were computed using the grey-scale form of Algorithm 1. The pixelwise minimum of these was then obtained, giving a final result which is still therefore a closing. In the first two instances, the adjacency graphs were those of Fig. 2(a) and (c)—that is, adjacencies directed towards the north and north-east respectively. The other two closings were those based on similar adjacencies directed towards the east and south-east.

The result of this combination was to choose a family of paths such that at each point the entirety of the path was contained in a 90 degree angle double-ended cone, either vertically or diagonally. In some sense this captures the idea of a family of oriented, but flexible structuring elements. It is of course possible to modify the adjacency graphs in order to constrain these paths more or less.

10. Conclusions

In this paper we have explored the theory of path openings and closings on binary and grey-level images. Path openings are openings over a large number of connected or disconnected paths, which extend the useful notions of openings by unions of line segments by

allowing the use of oriented, narrow but non-straight segments as a family of structuring elements. Because of the oriented nature of the family of structuring elements used, the resulting operators are more constrained than area openings.

Path openings and closings essentially allow practitioners to close the gap between openings by line segments (which are constrained and anisotropic) and area openings (which are unconstrained and isotropic). The framework developed in this paper allows for paths which behave more closely like one or the other, by varying the adjacency relation.

We have developed a workable solution for such path openings with low complexity, which makes the computation of such paths practical.

Finally we have explored the questions of how to deal with border effects, how to compute path opening transforms (only in the binary case) and how to extend path openings to incomplete paths, which could provide a degree of robustness against noise.

Note

1. 'Reflexive' would mean that $x \mapsto x$ for every $x \in E$. 'Symmetric' would mean that $x \mapsto y$ iff $y \mapsto x$, for every $x, y \in E$.

References

1. M. Buckley and H. Talbot, "Flexible linear openings and closings," in *Mathematical Morphology and its Application to Image Analysis*, L. Vincent and D. Bloomberg (Eds.), Palo Alto, ISMM, Kluwer, June 2000.
2. H.J.A.M. Heijmans, *Morphological Image Operators*, Academic Press: Boston, 1994.
3. H.J.A.M. Heijmans, "Morphological filters for dummies," in *Mathematical Morphology and its Applications to Image and Signal Processing*, Petros Maragos, Ronald W. Schafer, and Muhammad A. Butt (Eds.), Atlanta, GA, Proceedings for ISMM'96, Kluwer Acad., May 1996, pp. 127–137.
4. C. Ronse, "Extraction of narrow peaks and ridges in images by combination of local low rank and max filters: Implementation and applications to clinical angiography," Working Document WD47, Philips Research Laboratories, Brussels, Belgium, 1988.
5. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press: London, 1982.
6. J. Serra (Ed.), *Image Analysis and Mathematical Morphology. II: Theoretical Advances*, Academic Press: London, 1988.
7. P. Soille, *Morphological Image Analysis, Principles and Applications*, Springer, 1999.
8. P. Soille and H. Talbot, "Directional morphological filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 11, pp. 1313–1329, 2001.
9. M. Vandroogenboeck and H. Talbot, "Fast computation of morphological operations with arbitrary structuring elements," *Pattern Recognition Letters*, Vol. 17, pp. 1451–1460, 1996.
10. L. Vincent, "Minimal path algorithms for the robust detection of linear features in images," in *Mathematical Morphology and its Applications to Image and Signal processing*, vol. 12 of *Computational Imaging and Vision*, Amsterdam, Kluwer, Proceedings for ISMM'98, June 1998, pp. 331–338.

Henk Heijmans was born in 1957 in Tilburg, The Netherlands. After obtaining his masters degree in mathematics from the Technical University in Eindhoven, he went to the Centre for Mathematics and Computer Science where he became a PhD student working in the area of mathematical biology. He received his PhD degree from the University of Amsterdam in 1985. Currently Henk Heijmans is heading the "Signals and Images" research group (PNA4) at CWI, Amsterdam, The Netherlands. The aim of this research theme is to combine expertises in various directions in signal and image analysis, in which mathematics and computer science play an essential role, thereby contributing to solutions of important problems for society in the coming years. His own research interests are focused towards mathematical techniques for image and signal processing, with an emphasis on mathematical morphology and wavelet analysis.

Michael Buckley is a Principal Research Scientist at CSIRO, Mathematical and Information Sciences, with the Biotechnology and Health Informatics research program. He has a Ph.D. in Mathematical Statistics. In recent years Michael has published several papers on microarray normalisation. In 2003–2004 he has worked in proteomics, including protein chips, LC-MS-MS and biomarker discovery. He has substantial experience in dynamic programming, Hidden Markov Models and related methods for matching and modelling amino acid or nucleotide sequences. He has also worked in the field of image analysis, including the Road Crack project where he was the technical leader of algorithm development, collaborations with Axon Instruments Inc. and analysis of cDNA microarray images.

Hugues Talbot received the engineering degree from École Centrale de Paris in 1989, the D.E.A. (Masters) from University Paris VI in 1990 and the Ph.D from École des Mines de Paris in 1993, under the guidance of Dominique Jeulin and Jean Serra. He has been a scientist with the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Mathematical and Information Sciences since 1994. He has worked on numerous applied projects in relation with industry, has contributed more than 30 research papers in international journals and conferences and has co-edited two sets of international conference proceedings on image analysis. He was the general chair for DICTA'2003, the largest regular image processing conference in Australia. He now also teaches image processing at the University of Sydney, and his research interests include image segmentation, linear structure analysis, texture analysis and algorithms.

Appendix F

Joint Quantizing and Denoising

This article was published in the Journal of Mathematical Imaging and Vision as Chaux et al. (2010).


```
@article{springerlink:10.1007/s10851-010-0241-3,
Affiliation = {Lab. Informatique Gaspard Monge, UMR CNRS 8049,
Universit{\'e} Paris-Est, Champs-sur-Marne, 77454 Marne-la-Vall{\'e}e, France},
Author = {Chaux, Caroline and Jezierska, Anna and Pesquet,
Jean-Christophe and Talbot, Hugues},
Issn = {0924-9907},
Journal = {Journal of Mathematical Imaging and Vision},
Keyword = {Computer Science},
Note = {10.1007/s10851-010-0241-3},
Pages = {1-16},
Publisher = {Springer Netherlands},
Title = {A Spatial Regularization Approach for Vector Quantization},
Url = {http://dx.doi.org/10.1007/s10851-010-0241-3},
Year = {2010},
Bdsk-Url-1 = {http://dx.doi.org/10.1007/s10851-010-0241-3}}
```

J Math Imaging Vis (2011) 41:23–38
DOI 10.1007/s10851-010-0241-3

A Spatial Regularization Approach for Vector Quantization

Caroline Chaux · Anna Jezierska · Jean-Christophe Pesquet · Hugues Talbot

Published online: 1 December 2010

Abstract Quantization, defined as the act of attributing a finite number of levels to an image, is an essential task in image acquisition and coding. It is also intricately linked to image analysis tasks, such as denoising and segmentation. In this paper, we investigate vector quantization combined with regularity constraints, a little-studied area which is of interest, in particular, when quantizing in the presence of noise or other acquisition artifacts. We present an optimization approach to the problem involving a novel two-step, iterative, flexible, joint quantizing-regularization method featuring both convex and combinatorial optimization techniques. We show that when using a small number of levels, our approach can yield better quality images in terms of SNR, with lower entropy, than conventional optimal quantization methods.

Keywords Vector quantization · Convex optimization · Combinatorial optimization · Proximal methods · Graph cuts · Image coding · Compression · Information theory · Entropy · Segmentation · Denoising · Regularization

This work was supported by the Agence Nationale de la Recherche under grant ANR-09-EMER-004-03.

C. Chaux · A. Jezierska (✉) · J.-C. Pesquet · H. Talbot
Lab. Informatique Gaspard Monge, UMR CNRS 8049, Université Paris-Est, Champs-sur-Marne, 77454 Marne-la-Vallée, France
e-mail: anna.jezierska@univ-paris-est.fr

C. Chaux
e-mail: caroline.chaux@univ-paris-est.fr

J.-C. Pesquet
e-mail: jean-christophe.pesquet@univ-paris-est.fr

H. Talbot
e-mail: hugues.talbot@univ-paris-est.fr

1 Introduction

Quantization is a fundamental task in digital image processing and information theory [1]. It plays a prominent role in early processing stages such as image digitization, and it is essential in lossy coding. It bears close resemblance to high level tasks such as denoising, segmentation, and data classification. In particular, quantizing a grey scale image in Q levels can be viewed as a classification or segmentation of the image in Q areas following an intensity homogeneity criterion. Each segmented area then corresponds to a decision class of the quantizer.

A classical solution for designing an optimal quantizer of a monochrome image is provided by the celebrated Lloyd-Max (LM) algorithm [2, 3]. An extension to the general vector case is the Linde–Buzo–Gray (LBG) algorithm [4]. The LBG algorithm proceeds iteratively by alternatively optimizing codevectors and decision classes so as to minimize a flexible quantization error measure. It is known to present good convergence properties in practice [5, 6]. However, one drawback is the lack of spatial regularity of the quantized image. Spatially smooth properties may be useful in low-rate compression when using advanced coding algorithms (e.g based on run length, differential or multi-resolution techniques), especially in the context of medical and low bit-rate video compression applications like compression of confocal laser scanning microscopy image sequences [7] or mobile television [8]. It may also be of interest for quantizing images featuring noise. In the latter case, quantization can be viewed as a means for denoising discrete-valued images that are piecewise constant.

Since the LBG algorithm is closely related to K-means, which are widely used in data classification, a possibility to enforce spatial smoothness of the quantized image would be to resort to fuzzy C-means clustering techniques and their

extensions [9]. These algorithms are however based on local measures of smoothness. Furthermore, an interesting approach was proposed by Alvarez *et al.* [10]. However, this method is based on reaction-diffusion PDEs and it addresses the quantization of grey-scale images, while our approach is more general and applicable to multicomponent images.

In this paper, we propose a quantization method that enforces some global spatial smoothness. This is achieved by introducing an adjustable regularization term in the minimization criterion, in addition to a quantization error measure. Similarly to the LBG algorithm, the optimal design of the quantizer is performed iteratively by alternating the minimization of a label field $i_{\mathcal{P}}$ and of a codebook $\boldsymbol{r}$. The latter minimization reduces to a convex optimization problem whereas the former is carried out by efficient combinatorial optimization techniques.

Section 2 describes the background of the work and introduces the notation used throughout the paper. The considered regularization approach is formulated in Sect. 3. Section 4 describes the proposed quantizer design algorithm. Section 5 provides more details on the combinatorial optimization step. Finally, some simulation results are provided in Sect. 6 both for grey scale and color images to show the effectiveness of the proposed quantization method before a conclusion is drawn in Sect. 7.

2 Background

We consider the vector quantization of a multichannel image $f = (f(n,m))_{(n,m)\in\mathbb{V}}$ where $\mathbb{V} = \{1,\ldots,N\} \times \{1,\ldots,M\}$ is the image support and, for every $(n,m) \in \mathbb{V}$,

$$f(n,m) = \big(f_1(n,m),\ldots,f_D(n,m)\big)^{\top} \in \mathbb{R}^D. \tag{1}$$

A similar notation will be used for the D-channel fields defined throughout the paper. Example of such multivariate data are complex-valued images ($D = 2$), color images ($D = 3$), multispectral images (D usually less than 10), hyperspectral images (D usually more than 10), ... In the following, the vector quantizer will operate on each D-dimensional vector of pixel values. The case when $D = 1$ corresponds to a scalar quantization of a monochannel image.

In order to define such a vector quantizer, we introduce the following variables: Q is a stricly positive integer, $\mathcal{P} = (\mathbb{D}_k)_{1\leq k\leq Q}$ is a partition of $\mathbb{V}$ and $\boldsymbol{r} = (r_1,\ldots,r_Q)$ is a matrix belonging to a nonempty closed convex subset C of $\mathbb{R}^{D\times Q}$. The role of this constraint set will be made more explicit in the next sections. The partition $\mathcal{P}$ can be characterized by the label image $(i_{\mathcal{P}}(n,m))_{(n,m)\in\mathbb{V}} \in \{1,\ldots,Q\}^{N\times M}$, defined as: for every $(n,m) \in \mathbb{V}$ and $k \in \{1,\ldots,Q\}$,

$$i_{\mathcal{P}}(n,m) = k \quad \Leftrightarrow \quad (n,m) \in \mathbb{D}_k. \tag{2}$$

A vector quantized image over Q codevectors $r_1,\ldots,r_Q$ and associated with the partition $\mathcal{P}$ is then given by

$$q_{i_{\mathcal{P}},\boldsymbol{r}} = (r_{i_{\mathcal{P}}(n,m)})_{(n,m)\in\mathbb{V}} \in \{r_1,\ldots,r_Q\}^{N\times M}. \tag{3}$$

A numerical example is given below to better explain the relation between variables Q, $\boldsymbol{r}$, $i_{\mathcal{P}}$ and $q_{i_{\mathcal{P}},\boldsymbol{r}}$, which play a prominent role in the rest of the paper. For instance, if a quantization over 2 bits of a 3×3 monochannel image is performed, we have $N = M = 3$, $D = 1$, $Q = 4$, and we may have $\boldsymbol{r} = (1,4,9,10)$ and $i_{\mathcal{P}} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 4 & 1 \\ 3 & 2 & 3 \end{bmatrix}$, then $q_{i_{\mathcal{P}},\boldsymbol{r}} = \begin{bmatrix} 1 & 4 & 9 \\ 1 & 10 & 1 \\ 9 & 4 & 9 \end{bmatrix}$. Note that the $i_{\mathcal{P}}$ matrix values belong to the set $\{1,2,3,4\}$ which correspond to the set of labels and $q_{i_{\mathcal{P}},\boldsymbol{r}}$ matrix values belong to $\boldsymbol{r}$.

An "optimally" quantized image $q_{i_{\overline{\mathcal{P}}},\overline{\boldsymbol{r}}}$ of f is usually obtained by looking for $(i_{\overline{\mathcal{P}}},\overline{\boldsymbol{r}}) \in \{1,\ldots,Q\}^{N\times M}\times C$ solution to the following problem:

$$\underset{(i_{\mathcal{P}},\boldsymbol{r})\in\{1,\ldots,Q\}^{N\times M}\times C}{\text{minimize}} \ \Phi(q_{i_{\mathcal{P}},\boldsymbol{r}}, f) \tag{4}$$

where $\Phi\colon (\mathbb{R}^D)^{N\times M} \times (\mathbb{R}^D)^{N\times M} \to \,]-\infty,+\infty]$ is some measure of the quantization error.

Standard choices for Φ are separable functions of the form

$$\big(\forall g = (g(n,m))_{(n,m)\in\mathbb{V}} \in (\mathbb{R}^D)^{N\times M}\big)$$
$$\Phi(g,f) = \sum_{n=1}^{N}\sum_{m=1}^{M} \varphi_{n,m}\big(g(n,m), f(n,m)\big) \tag{5}$$

where, for every $(n,m) \in \{1,\ldots,N\} \times \{1,\ldots,M\}$, $\varphi_{n,m}\colon \mathbb{R}^D \times \mathbb{R}^D \to \,]-\infty,+\infty]$. For example, one can use:

– the matrix weighted quadratic norm

$$\varphi_{n,m}\big(g(n,m), f(n,m)\big) = \big\|g(n,m) - f(n,m)\big\|^2_{\Gamma_{n,m}} \tag{6}$$

where $\Gamma_{n,m} \in \mathbb{R}^{D\times D}$ is a symmetric definite positive matrix and we have used the notation

$$(\forall a \in \mathbb{R}^D) \quad \|a\|_{\Gamma_{n,m}} = (a^{\top}\Gamma_{n,m}a)^{1/2}; \tag{7}$$

– the weighted ℓ_p norm measure ($p \in [1,+\infty[$)

$$\varphi_{n,m}\big(g(n,m), f(n,m)\big) = \sum_{d=1}^{D} \omega_d(n,m)|g_d(n,m) - f_d(n,m)|^p \tag{8}$$

where $\omega_d(n,m) \in [0,+\infty[$. As a special case, a mean absolute error criterion is found when $p = 1$.

– the generalized Kullback-Leibler divergence

$$\varphi_{n,m}\big(g(n,m),\, f(n,m)\big) = \sum_{d=1}^{D} \kappa\big(g_d(n,m),\, f_d(n,m)\big) \quad (9)$$

where

$$\big(\forall (u,v) \in \mathbb{R}^2\big)$$

$$\kappa(u,v) = \begin{cases} -v\ln(u/v) + u - v \\ \quad \text{if } (u,v) \in \,]0,+\infty[^2, \\ u \quad \text{if } u \in [0,+\infty[\text{ and } v = 0, \\ +\infty \quad \text{otherwise.} \end{cases} \quad (10)$$

Maximum error measures may also be useful, which are expressed as

$$\big(\forall g = (g(n,m))_{(n,m)\in\mathbb{V}} \in (\mathbb{R}^D)^{N\times M}\big)$$

$$\Phi(g,f) = \max_{\substack{1\le n\le N \\ 1\le m\le M}} \varphi_{n,m}\big(g(n,m),\, f(n,m)\big) \quad (11)$$

where, for every $(n,m) \in \mathbb{V}$, $\varphi_{n,m}\colon \mathbb{R}^D \times \mathbb{R}^D \to \,]-\infty,+\infty]$. For example, we can use the sup norm:

$$\varphi_{n,m}\big(g(n,m),\, f(n,m)\big) = \max_{1\le d\le D} |g_d(n,m) - f_d(n,m)|. \quad (12)$$

In this context, a numerical solution to problem (4) when $C = \mathbb{R}^{D\times Q}$ is provided by the LBG algorithm, the general form of which is recalled below.

Algorithm 1 (LBG Algorithm)

Fix $Q \in \mathbb{N}^*$ and $\boldsymbol{r}^{(0)} \in \mathbb{R}^{D\times Q}$.
For $\ell = 0, 1, \ldots$
$\quad i_{\mathcal{P}}^{(\ell)} \in \operatorname{Argmin}_{i_{\mathcal{P}}\in\{1,\ldots,Q\}^{N\times M}} \Phi(q_{i_{\mathcal{P}},\boldsymbol{r}^{(\ell)}}, f)$
$\quad \boldsymbol{r}^{(\ell+1)} \in \operatorname{Argmin}_{\boldsymbol{r}\in\mathbb{R}^{D\times Q}} \Phi(q_{i_{\mathcal{P}}^{(\ell)},\boldsymbol{r}}, f)$

For separable and maximum error measures (see (5) and (11)), the optimization of the label field at iteration ℓ then amounts to applying a nearest neighbour rule, i.e. finding $i_{\mathcal{P}}^{(\ell)}$ such that, for every $(n,m) \in \mathbb{V}$ and, for every $k \in \{1,\ldots,Q\}$, $i_{\mathcal{P}}^{(\ell)}(n,m) = k$ only if

$$(\forall k' \in \{1,\ldots,Q\})$$

$$\varphi_{n,m}\big(r_k,\, f(n,m)\big) \le \varphi_{n,m}\big(r_{k'},\, f(n,m)\big). \quad (13)$$

Note that, in general, $i_{\mathcal{P}}^{(\ell)}(n,m)$ is not uniquely defined since there may exist $k' \in \{1,\ldots,Q\} \setminus \{k\}$ such that $\varphi_{n,m}(r_k, f(n,m)) = \varphi_{n,m}(r_{k'}, f(n,m))$.

On the other hand, updating of the codebook at iteration ℓ is performed by computing the centroid of each region $\mathbb{D}_k^{(\ell)}$, $k \in \{1,\ldots,Q\}$. For the matrix weighted quadratic norm ((5) and (6)), we thus obtain the classical center of mass of $\mathbb{D}_k^{(\ell)}$:

$$r_k^{(\ell+1)} = \Big(\sum_{(n,m)\in\mathbb{D}_k^{(\ell)}} \Gamma_{n,m}\Big)^{-1}\Big(\sum_{(n,m)\in\mathbb{D}_k^{(\ell)}} \Gamma_{n,m} f(n,m)\Big). \quad (14)$$

For the mean absolute value criterion ((5) and (8) with $p = 1$ and equal weights), $r_k^{(\ell+1)}$ is the vector median of the pixel values located in $\mathbb{D}_k^{(\ell)}$:

$$r_k^{(\ell+1)} = \big(\operatorname{median}\big\{f_d(n,m) \;\big|\; (n,m) \in \mathbb{D}_k^{(\ell)}\big\}\big)_{1\le d\le D}. \quad (15)$$

For the generalized Kullback-Leibler divergence ((5), (9) and (10)), we get

$$r_k^{(\ell+1)} = \frac{1}{\operatorname{card}\mathbb{D}_k^{(\ell)}} \sum_{(n,m)\in\mathbb{D}_k^{(\ell)}} f(n,m) \quad (16)$$

provided that $f \in ([0,+\infty[^D)^{N\times M}$. For the sup norm ((11) and (12)), we have

$$r_k^{(\ell+1)} = \Big(\frac{\beta_{d,k} + \gamma_{d,k}}{2}\Big)_{1\le d\le D} \quad (17)$$

where $\beta_{d,k} = \min\{f_d(n,m) | (n,m) \in \mathbb{D}_k^{(\ell)}\}$ and $\gamma_{d,k} = \max\{f_d(n,m) | (n,m) \in \mathbb{D}_k^{(\ell)}\}$.

When a closed form expression of $r_k^{(\ell+1)}$ is not available, one may resort to numerical optimization algorithms [11] to compute centroids.

It can also be noticed that an alternative to the LBG algorithm is the dynamic programming approach proposed in [12] (see also [13, 14] for more recent extensions) which features better global convergence properties. Generally, if LBG used, the final solution is sub-optimal.

3 Considered Design Criterion

One drawback of the approach described in the previous section is that it does not guarantee any spatial homogeneity of the resulting quantized image. To alleviate this shortcoming, we propose to solve the following problem:

$$\underset{(i_{\mathcal{P}},\boldsymbol{r})\in\{1,\ldots,Q\}^{N\times M}\times C}{\text{minimize}} \Phi(q_{i_{\mathcal{P}},\boldsymbol{r}}, f) + \rho(i_{\mathcal{P}}) \quad (18)$$

where $\rho\colon \{1,\ldots,Q\}^{N\times M} \to \,]-\infty,+\infty]$ is some penalty function which is used to promote the spatial regularity of the label image. Note that an alternative approach for ensuring the smoothness of the quantized image would be to solve a problem of the form

$$\underset{(i_{\mathcal{P}},\boldsymbol{r})\in\{1,\ldots,Q\}^{N\times M}\times C}{\text{minimize}} \Phi(q_{i_{\mathcal{P}},\boldsymbol{r}}, f) + \widetilde{\rho}(q_{i_{\mathcal{P}},\boldsymbol{r}}) \quad (19)$$

where the regularization term $\widetilde{\rho}$ is now a function from $(\mathbb{R}^D)^{N\times M}$ to $]-\infty,+\infty]$. The latter problem appears however more difficult to solve than (18) since the regularization term in (19) is a multivariate function depending both on $i_{\mathcal{P}}$ and $\boldsymbol{r}$.

The existence of a solution to problem (18) is secured by the following result:

Proposition 1 *Assume that $\Phi(\cdot, f)$ is a lower-semicontinuous function and that one of the following conditions holds*:

(i) $\Phi(\cdot, f)$ *is coercive*;[1]
(ii) C *is bounded*.

Then, problem (18) *has a solution*.

Proof Let $i_{\mathcal{P}}$ be any given label field in $\{1,\ldots,Q\}^{N\times M}$. According to (3), $\boldsymbol{r}\mapsto q_{i_{\mathcal{P}},\boldsymbol{r}}$ is a linear operator, and consequently $\boldsymbol{r}\mapsto\Phi(q_{i_{\mathcal{P}},\boldsymbol{r}}, f)$ is a lower-semicontinuous function. As a direct consequence of Weierstrass theorem [15], under Assumption (i) or (ii), there exists $\overline{\boldsymbol{r}}_{\mathcal{P}}\in C$ such that

$$\Phi(q_{i_{\mathcal{P}},\overline{\boldsymbol{r}}_{\mathcal{P}}}, f)=\min_{\boldsymbol{r}\in C}\Phi(q_{i_{\mathcal{P}},\boldsymbol{r}}, f). \tag{20}$$

Problem (18) can thus be reexpressed as

$$\underset{i_{\mathcal{P}}\in\{1,\ldots,Q\}^{N\times M}}{\text{minimize}}\ \Phi(q_{i_{\mathcal{P}},\overline{\boldsymbol{r}}_{\mathcal{P}}}, f)+\rho(i_{\mathcal{P}}). \tag{21}$$

The latter minimization can be performed by a search among a finite number of candidate values, so leading to an optimal label field $i_{\overline{\mathcal{P}}}$. Hence, $(i_{\overline{\mathcal{P}}}, q_{i_{\overline{\mathcal{P}}},\overline{\boldsymbol{r}}_{\overline{\mathcal{P}}}})$ is a solution to problem (18). □

Typical choices for ρ in (18) that can be made are the following:

– isotropic variation functions

$$\rho(i_{\mathcal{P}})=\mu\sum_{n=1}^{N-1}\sum_{m=1}^{M-1}\psi(\|\nabla i_{\mathcal{P}}(n,m)\|),\quad \mu\geq 0 \tag{22}$$

where $\nabla i_{\mathcal{P}}(n,m)=(i_{\mathcal{P}}(n+1,m)-i_{\mathcal{P}}(n,m), i_{\mathcal{P}}(n,m+1)-i_{\mathcal{P}}(n,m))$ is the discrete gradient of $i_{\mathcal{P}}$ at location (n,m).

– anisotropic variation functions

$$\rho(i_{\mathcal{P}})=\mu\left(\sum_{n=1}^{N-1}\sum_{m=1}^{M}\psi(|i_{\mathcal{P}}(n+1,m)-i_{\mathcal{P}}(n,m)|)+\sum_{n=1}^{N}\sum_{m=1}^{M-1}\psi(|i_{\mathcal{P}}(n,m+1)-i_{\mathcal{P}}(n,m)|)\right),\quad \mu\geq 0. \tag{23}$$

[1] This means that $\lim_{\|g\|\to+\infty}\Phi(g, f)=+\infty$.

In the above two examples, ψ is a function from $[0,+\infty[$ to $]-\infty,+\infty]$. When ψ is the identity function, the classical isotropic or anisotropic total variations are obtained. A more flexible form is given by the truncated linear function [16] defined as

$$(\forall x\in[0,+\infty[)\quad \psi(x)=\begin{cases}x & \text{if } x<\zeta,\\ \zeta & \text{otherwise}\end{cases} \tag{24}$$

where $\zeta>0$ is the limiting constant. If $\psi=(\cdot)^2$, then a Tikhonov-like regularization is performed. Another interesting choice of ψ is the binary cost function (also named ℓ_0 criterion).

$$(\forall x\in[0,+\infty[)\quad \psi(x)=\begin{cases}0 & \text{if } x=0,\\ 1 & \text{otherwise.}\end{cases} \tag{25}$$

When ψ is a (strictly) increasing function, higher local differences of the label values entail a stronger penalization. For this behaviour to be consistent with the quantized image values, some ordering relation should typically exist between the codevectors. Hence, if $D=1$, a natural choice is to constrain the vector $\boldsymbol{r}$ to belong to the closed convex cone:

$$C=\left\{(s_1,\ldots,s_Q)\in\mathbb{R}^Q\;\middle|\; s_1\leq\cdots\leq s_Q\right\}. \tag{26}$$

When $D>1$, the definition of C becomes more debatable since there exists no total order on $\mathbb{R}^D$. A possibility is to impose an artificial total order. In mathematical morphology, authors have proposed various lexicographic orderings [17, 18] or bit-mixing [19] along space-filling (Peano-like) curves.

A possible choice for C is the closed convex cone:

$$C=\left\{(s_1,\ldots,s_Q)\in\mathbb{R}^{D\times Q}\;\middle|\;\theta(s_1)\leq\cdots\leq\theta(s_Q)\right\} \tag{27}$$

where

$$(\forall u\in\mathbb{R}^D)\quad \theta(u)=\eta^{\top}u \tag{28}$$

and $\eta\in\mathbb{R}^D$. For example, for color images, by an appropriate choice of $\eta\in\mathbb{R}^3$ (possibly depending on the considered color system [20]), the function θ may serve to extract the luminance component of the codevectors.

More generally, the parameter vector $\eta\in\mathbb{R}^D$ may be obtained through a principal component analysis [21] of the original multichannel data. Note that, when the binary function in (25) is employed, the magnitudes of the local differences of the label fields have no influence as soon as they are nonzero. This means that ordering the codevectors does not appear useful in this case, and that one can set $C=\mathbb{R}^{D\times Q}$.

In addition to these considerations, when the regularization constant μ in (22) or (23) takes large values, solving (18) under the constraints modeled by (26) may lead to very close or even equal values of codevectors. As a consequence, the readibility of the quantized image may be affected. In

some applications, it may therefore be beneficial to redefine the constraint C in order to prevent this effect. When $D = 1$, the closed convex set C can thus be given by

$$C = \big\{(s_1, \ldots, s_Q) \in \mathbb{R}^Q \;\big|\; (\forall k \in \{1, \ldots, Q-1\})\; s_{k+1} - s_k \geq \delta\big\} \tag{29}$$

where $\delta \geq 0$. Similarly, when $D > 1$, we propose to set

$$C = \big\{(s_1, \ldots, s_Q) \in \mathbb{R}^{D\times Q} \;\big|\; (\forall k \in \{1, \ldots, Q-1\})\; \theta(s_{k+1} - s_k) \geq \delta\big\} \tag{30}$$

where $\delta \geq 0$ and θ is the function given by (28). Penalization of quantization values for being too close to each other was previously introduced in the energy model proposed by Alvarez *et al.* [10].

4 Proposed Optimization Method

Even if $\Phi(\cdot, f)$ and ρ are convex functions, problem (18) is a nonconvex optimization problem due to the fact that $i_{\mathcal{P}}$ belongs to a (nonconvex) set of discrete values. In order to solve numerically this problem, we propose to use the following alternating optimization algorithm:

Algorithm 2 (Proposed algorithm)

Fix $Q \in \mathbb{N}^*$ and $\boldsymbol{r}^{(0)} \in C$.
For $\ell = 0, 1, \ldots$
$\quad i_{\mathcal{P}}^{(\ell)} \in \operatorname{Argmin}_{i_{\mathcal{P}} \in \{1,\ldots,Q\}^{N\times M}} \Phi(q_{i_{\mathcal{P}}, \boldsymbol{r}^{(\ell)}}, f) + \rho(i_{\mathcal{P}})$
$\quad \boldsymbol{r}^{(\ell+1)} \in \operatorname{Argmin}_{\boldsymbol{r}\in C} \Phi(q_{i_{\mathcal{P}}^{(\ell)}, \boldsymbol{r}}, f)$

It is worth noticing that this algorithm constitutes an extension of the LBG algorithm (see Algorithm 1) which would correspond to the case when ρ is the null function and $C = \mathbb{R}^{D\times Q}$. Similarly to the LBG algorithm, under the assumptions of Proposition 1, Algorithm 2 generates a sequence $(i_{\mathcal{P}}^{(\ell)}, \boldsymbol{r}^{(\ell+1)})_{\ell\in\mathbb{N}}$ such that $(\Phi(q_{i_{\mathcal{P}}^{(\ell)}, \boldsymbol{r}^{(\ell+1)}}, f) + \rho(i_{\mathcal{P}}^{(\ell)}))_{\ell\in\mathbb{N}}$ is a convergent decaying sequence. At each iteration ℓ, the determination of $i_{\mathcal{P}}^{(\ell)}$ given $\boldsymbol{r}^{(\ell)}$ is a combinatorial optimization problem for which there exist efficient solutions for particular choices of Φ and ρ, as explained in the next section.

In turn, if $\Phi(\cdot, f)$ is a convex function, the determination of $\boldsymbol{r}^{(\ell+1)}$ given $i_{\mathcal{P}}^{(\ell)}$ is a constrained convex optimization problem the solution of which can be determined numerically. For any given $i_{\mathcal{P}} \in \{1, \ldots, Q\}^{N\times M}$, let $L_{i_{\mathcal{P}}}$ be the linear operator defined as

$$L_{i_{\mathcal{P}}}: \mathbb{R}^{D\times Q} \to (\mathbb{R}^D)^{N\times M},$$
$$\boldsymbol{r} \mapsto q_{i_{\mathcal{P}}, \boldsymbol{r}} \tag{31}$$

the adjoint of which is

$$L_{i_{\mathcal{P}}}^*: (\mathbb{R}^D)^{N\times M} \to \mathbb{R}^{D\times Q},$$
$$g \mapsto \Big(\sum_{(n,m)\in\mathbb{D}_1} g(n,m), \ldots, \sum_{(n,m)\in\mathbb{D}_Q} g(n,m)\Big) \tag{32}$$

(with the convention $\sum_{(n,m)\in\varnothing} \cdot = 0$). Then,

$$L_{i_{\mathcal{P}}}^* L_{i_{\mathcal{P}}}: \mathbb{R}^{D\times Q} \to \mathbb{R}^{D\times Q},$$
$$\boldsymbol{r} \mapsto \boldsymbol{r}\,\operatorname{Diag}(\operatorname{card}\mathbb{D}_1, \ldots, \operatorname{card}\mathbb{D}_Q). \tag{33}$$

In addition, let Θ be the linear operator defined as

$$\Theta: \mathbb{R}^{D\times Q} \to \mathbb{R}^{Q-1}, \tag{34}$$
$$(s_1, \ldots, s_Q) \mapsto \big(\theta(s_2 - s_1), \ldots, \theta(s_Q - s_{Q-1})\big) \tag{35}$$

where θ is given by (28) (with $\eta = 1$ when $D = 1$). The set C defined in (29) or (30) is thus equal to $\Theta^{-1}([\delta, +\infty[^{Q-1})$. Hence, the problem of minimization of $\boldsymbol{r} \mapsto \Phi(q_{i_{\mathcal{P}}, \boldsymbol{r}}, f)$ over C can be reexpressed as

$$\underset{\boldsymbol{r}\in\mathbb{R}^{D\times Q}}{\text{minimize}}\; \Phi(L_{i_{\mathcal{P}}}\boldsymbol{r}, f) + \iota_{[\delta,+\infty[^{Q-1}}(\Theta\boldsymbol{r}) \tag{36}$$

where ι_S denotes the indicator function of a set S, which is zero on S and equal to $+\infty$ on its complement. If we assume that $\Phi(\cdot, f)$ belongs to $\Gamma_0((\mathbb{R}^D)^{N\times M})$, the class of lower-semicontinuous proper convex functions from $(\mathbb{R}^D)^{N\times M}$ to $]-\infty, +\infty]$, (36) can be solved through existing convex optimization approaches [11, 22, 23]. One possible solution is to employ the method proposed in [24] (hereafter called PPXA+) which constitutes an extension of the parallel proximal algorithm (PPXA) developed in [25] and of the simultaneous direction of multipliers method proposed in [26] (see also [27–29]).

Algorithm 3 (PPXA+ for solving (36))

Initialization
$\quad (\omega_1, \omega_2, \omega_3) \in]0, +\infty[^3$
$\quad t^{(1,0)} \in (\mathbb{R}^D)^{N\times M}, t^{(2,0)} \in \mathbb{R}^{Q-1}, \boldsymbol{s}^{(0)} \in \mathbb{R}^{D\times Q}$
$\quad R = (\omega_1 L_{i_{\mathcal{P}}}^* L_{i_{\mathcal{P}}} + \omega_2 \Theta^*\Theta + \omega_3 I)^{-1}$
$\quad \boldsymbol{r}^{(0)} = R\,(\omega_1 L_{i_{\mathcal{P}}}^* t^{(1,0)} + \omega_2 \Theta^* t^{(2,0)} + \omega_3 \boldsymbol{s}^{(0)})$
For $\ell = 0, 1, \ldots$
$\quad p^{(1,\ell)} = \operatorname{prox}_{\frac{1}{\omega_1}\Phi(\cdot,f)}\big(t^{(1,\ell)}\big)$
$\quad p^{(2,\ell)} = P_{[\delta,+\infty[^{Q-1}}\big(t^{(2,\ell)}\big)$
$\quad \boldsymbol{c}^{(\ell)} = R\,(\omega_1 L_{i_{\mathcal{P}}}^* p^{(1,\ell)} + \omega_2 \Theta^* p^{(2,\ell)} + \omega_3 \boldsymbol{s}^{(\ell)})$
$\quad \lambda_\ell \in]0, 2[$
$\quad t^{(1,\ell+1)} = t^{(1,\ell)} + \lambda_\ell\big(L_{i_{\mathcal{P}}}(2\boldsymbol{c}^{(\ell)} - \boldsymbol{r}^{(\ell)}) - p^{(1,\ell)}\big)$
$\quad t^{(2,\ell+1)} = t^{(2,\ell)} + \lambda_\ell\big(\Theta(2\boldsymbol{c}^{(\ell)} - \boldsymbol{r}^{(\ell)}) - p^{(2,\ell)}\big)$
$\quad \boldsymbol{s}^{(\ell+1)} = \boldsymbol{s}^{(\ell)} + \lambda_\ell(2\boldsymbol{c}^{(\ell)} - \boldsymbol{r}^{(\ell)} - \boldsymbol{s}^{(\ell)})$
$\quad \boldsymbol{r}^{(\ell+1)} = \boldsymbol{r}^{(\ell)} + \lambda_\ell(\boldsymbol{c}^{(\ell)} - \boldsymbol{r}^{(\ell)})$.

In the above algorithm, $\operatorname{prox}_{\frac{1}{\omega_1}\Phi(\cdot,f)}$ is the proximity operator of $\omega_1^{-1}\Phi(\cdot,f)$ [30] and $P_{[\delta,+\infty[^{Q-1}}$ is the projector onto $[\delta,+\infty[^{Q-1}$. Expressions of proximity operators for usual convex functions are listed in [31]. The convergence of the PPXA+ algorithm is guaranteed under weak assumptions.

Proposition 2 *Assume that*

(i) *there exists* $\underline{\lambda} \in]0,2[$ *such that* $(\forall \ell \in \mathbb{N})\ \underline{\lambda} \leq \lambda_{\ell+1} \leq \lambda_\ell$.
(ii) *There exists* $\overline{\boldsymbol{r}} \in \mathbb{R}^{D\times Q}$ *such that*

$$L_{i_{\mathcal{P}}}\overline{\boldsymbol{r}} \in \operatorname{ri}\operatorname{dom}\Phi(\cdot,f) \quad \text{and} \quad \Theta\overline{\boldsymbol{r}} \in]\delta,+\infty[^{Q-1} \tag{37}$$

where $\operatorname{dom}\Phi(\cdot,f)$ *is the domain of* $\Phi(\cdot,f)$ *and* $\operatorname{ri}\operatorname{dom}\Phi(\cdot,f)$ *is its relative interior.*

Then, the sequence $(\boldsymbol{r}^{(\ell)})_{\ell\in\mathbb{N}}$ *generated by Algorithm 3 converges to a solution to problem* (36).

Proof See [24]. □

5 Combinatorial Partitioning

We now consider two combinatorial optimization methods for finding

$$i_{\widehat{\mathcal{P}}} \in \operatorname*{Argmin}_{i_{\mathcal{P}}\in\{1,\ldots,Q\}^{N\times M}} \Phi(q_{i_{\mathcal{P}},\boldsymbol{r}}, f) + \rho(i_{\mathcal{P}}) \tag{38}$$

for a given value of $\boldsymbol{r} \in C$. Here we seek to use standard methods in combinatorial optimization which have proved to be useful in applications to image processing. In this context, a common form for regularization problems is the following:

$$\operatorname*{minimize}_{i_{\mathcal{P}}\in\{1,\ldots,Q\}^{N\times M}} \widetilde{\Phi}(i_{\mathcal{P}}, f) + \rho(i_{\mathcal{P}}), \tag{39}$$

where $\widetilde{\Phi}\colon \{1,\ldots,Q\}^{N\times M} \times (\mathbb{R}^D)^{N\times M} \to]-\infty,+\infty]$ is a data fidelity function, ρ a regularization function, f the initial image and $i_{\mathcal{P}}$ the target discrete one. To formulate our problem in this framework, we need to introduce the auxiliary function

$$\chi_{\boldsymbol{r}}\colon \{1,\ldots,Q\}^{N\times M} \mapsto \{r_1,\ldots,r_Q\}^{N\times M},$$
$$i_{\mathcal{P}} \mapsto q_{i_{\mathcal{P}},\boldsymbol{r}}.$$

Then, our problem becomes

$$\operatorname*{minimize}_{i_{\mathcal{P}}\in\{1,\ldots,Q\}^{N\times M}} \Phi(\chi_{\boldsymbol{r}}(i_{\mathcal{P}}), f) + \rho(i_{\mathcal{P}}). \tag{40}$$

Note that $\chi_{\boldsymbol{r}}$ is monotonic but nonlinear. Note further that the set $\{r_1,\ldots,r_Q\}$ changes at each iteration of the complete algorithm. However, during the regularization step, this set is fixed.

In this section, we use graph-cut based algorithms, which have proved to be effective in the context of smoothing, denoising and segmentation [32].

5.1 Method I: Convex Regularization Term

Here we describe a way to formulate the problem as a globally optimal graph cut, inspired by the approach of Ishikawa *et al.* [33]. In this approach, we build a discrete graph that will allow us to represent the quantized and regularized version of our original image. Let us define the oriented, edge-weighted graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ as follows:

(i) $\mathcal{V} = \mathbb{V}\times\{1,\ldots,Q\}\cup\{s,t\}$ the set of vertices quantized over Q levels, where $\mathbb{V}$ is the image support as defined in Sect. 2. We add two special vertices, the *source* s and the *sink* t.
(ii) $\mathcal{E} = \mathcal{E}_D \cup \mathcal{E}_C \cup \mathcal{E}_P$, the set of edges. In the following we denote an oriented edge by $[a,b]$, with a and b the vertices it joins in the direction from a to b. We have:
  (a) $\mathcal{E}_D = \bigcup_{v\in\mathbb{V}} \mathcal{E}_D^v$ the upward columns of the graph. For all $v \in \mathbb{V}$, let $h_{v,k}$ denote the node in column v and row k. A single column associated with pixel v is defined as

$$\mathcal{E}_D^v = \big\{[s,h_{v,1}]\big\} \cup \big\{[h_{v,k},h_{v,k+1}] \,\big|\, k\in\{1,\ldots,Q-1\}\big\} \cup \big\{h_{v,Q},t\big\},$$

  (b) $\mathcal{E}_C = \bigcup_{v\in\mathbb{V}} \mathcal{E}_C^v$ the downward columns of the graph, with

$$\mathcal{E}_C^v = \big\{[h_{v,k},h_{v,k+1}] \,\big|\, k\in\{1,\ldots,Q-1\}\big\},$$

  (c) and the penalty edges of the graph are thus defined as

$$\mathcal{E}_P = \big\{[h_{v,k},h_{w,k}] \,\big|\, \{v,w\} \text{ neighbours in } \mathbb{V},\ k\in\{1,\ldots,Q\}\big\}.$$

The above graph is depicted in Fig. 1. In this figure, for simplicity we assume each pixel has only two neighbours, which allows us to represent the graph in a 2D planar layout. For actual 2D images, there exist many more penalty edges between all neighbours in $\mathbb{V}$. The graph layout is then non-planar, but remains similar. For 2D images, it is best to see the arrangement of v vertices as in the original images, with the column of penalty edges in an extra dimension.

If Φ is the separable function defined in (5) and ρ is the anisotropic TV in (23) where ψ is the identity function, we define the capacities (or weights) c of edges $[a,b] \in \mathcal{E}$ as follows:

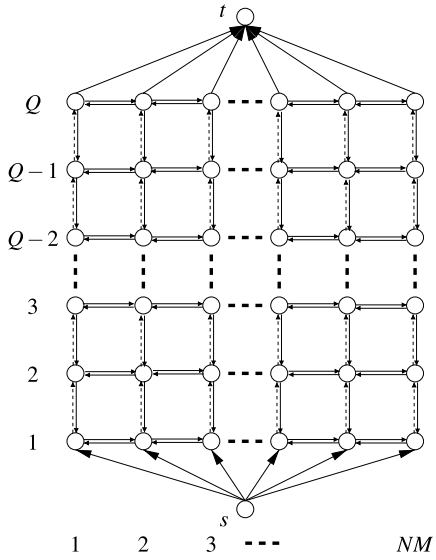


Fig. 1 Construction of the Ishikawa-like optimization graph. Arrows represent the edges $\mathcal{E}$ and circles the nodes in $\mathcal{V}$. Horizontal edges are in $\mathcal{E}_P$, the dotted upward vertical edges are in $\mathcal{E}_D$ and the plain downward vertical edges are in $\mathcal{E}_C$. Vertices s and t are respectively the source and the sink. All pixels in the image from 1 to NM are represented in the columns. In actual 2D images, there exist many more penalty edges $\mathcal{E}_P$ than depicted here: all those between neighbours in $\mathbb{V}$

(i) Links to the source have infinite capacity:

$$\forall v \in \mathbb{V}, \quad c\left([s, h_{v,1}]\right) = +\infty.$$

(ii) Data fidelity term for any pixel $v \in \mathbb{V}$ is

$$\forall k \in \{1, \ldots, Q-1\},$$
$$c\left([h_{v,k}, h_{v,k+1}]\right) = \varphi_v\big(r_k, f(v)\big),$$
$$c\left([h_{v,Q}, t]\right) = \varphi_v\big(r_Q, f(v)\big).$$

(iii) The capacity of downward columns is infinite to constrain a single cut per column:

$$\forall v \in \mathbb{V}, \forall k \in \{1 \ldots Q-1\},$$
$$c\left([h_{v,k+1}, h_{v,k}]\right) = +\infty.$$

(iv) The regularization term along the penalty edges of the graph is:

$$\text{for every } \{v, w\} \text{ neighbours in } \mathbb{V}, \forall k \in \{1, \ldots, Q\},$$
$$c\left([h_{v,k}, h_{w,k}]\right) = \mu$$

The above graph $\mathcal{G}$ has the same topology as the one proposed by Ishikawa and it can be extended to any convex function ψ [33]. The capacities of $\mathcal{E}$ are adjusted in such a way that a cut of $\mathcal{G}$ corresponds to the solution of (40), granted by the following result:

Proposition 3 *If ρ is the anisotropic TV in (23) where ψ is the identity function, then the min cut of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the globally optimal solution to (40).*

Proof This result is derived from the construction of the graph. First note that we build here a binary flow network with one source and one sink. Following Ishikawa, relying on the celebrated discrete maxflow/mincut theorem of Ford and Fulkerson [34], any binary cut that separates s and t along a series of edges, can be interpreted as a solution $i_{\mathcal{P}}$. Indeed, the infinite capacity of the downward edges ensure a single cut edge in each column of the graph, and the infinite capacity of the upward $[s, h_{v,1}]$ edges for all v ensures that, in all columns, this cut will be located above one of the nodes corresponding to a level $k \in \{1, \ldots, Q\}$. We can therefore associate the cut in column v with the value of the level immediately below the cut, and associate this with $i_{\mathcal{P}}(v)$. Recalling that all labels below the cut will have the same label as s, and all that above the cut the same label as t, the value of $i_{\mathcal{P}}$ at pixel v is the highest level l in column v of the graph that is labelled like the source s. Here, by convention, the source is labelled with 1 and the sink with 0. We can then write $i_{\mathcal{P}}(v) = \max\{k \mid h_{v,k} = 1\}$.

Now, the computation of the maxflow/mincut on this graph minimizes the energy of the cut, interpreted as the sum of two terms:

(i) since the downward constraint edges ensure a single cut edge along each column of the graph, this corresponds to contribution of the data fidelity term $\varphi_v(r_Q, f(v))$ to the total energy.
(ii) Similarly, we note that each penalty edges in $\mathcal{E}_P$ with capacity μ can be cut at most once. Let u and v be two neighbouring pixels in the graph. The cut along penalty edges between $i_{\mathcal{P}}(u)$ and $i_{\mathcal{P}}(v)$ crosses exactly as many penalty edges as there are quantization level differences between u and v. We note that this correspond to a contribution of $\mu|i_{\mathcal{P}}(u) - i_{\mathcal{P}}(v)|$ to the total energy.

Hence, the computation of the maxflow/mincut on this graph solves (40) exactly, in the case of (23), when ψ is the identity. □

Remark 1

(i) It is also possible to solve this problem exactly in the case when ψ is convex and not necessarily the identity, by adding non-horizontal penalty edges [35], but we do not consider this case here, as $\psi = \mathrm{Id}$ is favorable when discontinuities exist in the original image.
(ii) In the case when the number of quantized levels Q is small (say between 1 and 32), the Ishikawa framework is very efficient.
(iii) As the dimensionality of the problem increases, so does the number of penalty edges in the graph. The cut is always an hypersurface of codimension 1.

(iv) Ishikawa recommends solving the maxflow/mincut by using a push-relabel algorithm, which makes perfect sense as the dimensionality increases, because these algorithms have an asymptotic complexity independent of the number of edges.

5.2 Method II: Submodular Regularization Term

Since the method proposed in Sect. 5.1 works only for a convex function ψ, we propose to solve the general problem defined in (38) with the α-expansion algorithm [32], which has been proven to be very effective for some non-convex functions ψ such as the Potts model of (25). Though only a local minimum is then guaranteed, the resulting energy will be within a known factor of the global minimum energy [32]. Here we reintroduce the standard notation of α-expansions as we need to specify the capacities on the corresponding edges in the context of this article. Following Kolmogorov *et al.* [36], we build a directed graph for each quantization level, called α-expansion graph $\mathcal{G}_\alpha = (\mathcal{V}, \mathcal{E})$, defined as follows:

(i) $\mathcal{V} = V \cup \{\alpha, \overline{\alpha}\}$ is the set of vertices, with α and $\overline{\alpha}$ two special term nodes and $V = \{1, \ldots, NM\}$ is the set of image nodes;

(ii) $\mathcal{E} = \mathcal{E}_V \cup \mathcal{E}_\mathcal{N}$ is the set of edges, defined as follows:
   (a) $\mathcal{E}_V = \bigcup_{v \in V} \{[\alpha, v], [v, \overline{\alpha}]\}$ is the set of edges between special term nodes and image nodes;
   (b) $\mathcal{E}_\mathcal{N} = \bigcup_{\{u,v\} \text{ neighbours}}$ is the set of edges between neighbour and $\mathcal{N}$ is the set of neighbours pairs containing only ordered *pairs*$(u, v) \in V^2$, i.e. such that $u < v$.
   (c) The capacity for all edges are given in Table 1.

Computing the max-flow/min-cost cut of $\mathcal{G}_\alpha$ separates vertices α and $\overline{\alpha}$ in such a way that the α region can only expand, hence the name of the algorithm. The value of the function associating new values to $i_\mathcal{P}$, based on cut of $\mathcal{G}_\alpha$, is called "*α-move of $i_\mathcal{P}$*" [16]. The algorithm is as follows:

Algorithm 4 (α-expansion algorithm)

Fix $i_\mathcal{P}^{(0)}$
For $\ell = 0, 1, \ldots$
$\quad \widetilde{\alpha}^{(\ell)} \in \operatorname{Argmin}_{\alpha \in \{1,\ldots,Q\}} \{\Phi(\chi_{\boldsymbol{r}}(\widehat{i}_\mathcal{P}), f) + \rho(\widehat{i}_\mathcal{P}) \mid$
$\quad \widehat{i}_\mathcal{P} = \alpha\text{-move of } i_\mathcal{P}^{(\ell)}\}$
$\quad i_\mathcal{P}^{(\ell+1)} = \widetilde{\alpha}^{(\ell)}\text{-move of } i_\mathcal{P}^{(\ell)}$

Proposition 4 *If* (38) *is submodular then it can be solved with the α-expansion algorithm.*

Proof It is shown in [36] that in order to employ the α-expansion algorithm, (38) has to satisfy the following conditions at iteration ℓ:

(i) (38) has a binary representation of the form:

$$\text{minimize} \sum_{u \in V} B_1^{(\ell)}(b_{(n_u,m_u)}) + \sum_{\{u,v\} \text{ neighbours}} B_2^{(\ell)}(b_{(n_u,m_u)}, b_{(n_v,m_v)}), \tag{41}$$

where b is a binary field while $B_1^{(\ell)}$ and $B_2^{(\ell)}$ have binary arguments.

(ii) The binary representation b is graph-representable, which can be verified by testing if term $B_2^{(\ell)}$ satisfies the submodular inequality:

$$B_2^{(\ell)}(0,0) + B_2^{(\ell)}(1,1) \leq B_2^{(\ell)}(1,0) + B_2^{(\ell)}(0,1)\,. \tag{42}$$

We now propose the following binary formulation of (38) by defining:

$$B_1^{(\ell)}(b_{(n_u,m_u)}) = \varphi_{n_u,m_u}(r_{\widehat{i}_\mathcal{P}(n_u,m_u)}, f(n_u,m_u)) \tag{43}$$

and

$$B_2^{(\ell)}(b_{(n_u,m_u)}, b_{(n_v,m_v)}) = \psi(|\widehat{i}_\mathcal{P}(n_u,m_u) - \widehat{i}_\mathcal{P}(n_v,m_v)|) \tag{44}$$

where

$$\widehat{i}_\mathcal{P}(n_u,m_u) = \begin{cases} i_\mathcal{P}^{(\ell)}(n_u,m_u) & \text{if } b_{(n_u,m_u)} = 0, \\ \alpha & \text{if } b_{(n_u,m_u)} = 1. \end{cases} \tag{45}$$

More standard graph-cut formulations would only allow us to optimize (39). These formulations would be problematic because we would not be able to separate the two steps in the inner loop of Algorithm 2, and therefore no convergence property could be derived.

Assuming (38) submodular, then the terms of its binary representation defined in (44) satisfy (42). Furthermore, it is shown in [16] that for ψ defined as Potts model of (25) or the truncated linear function in (24), and when ρ is the anisotropic TV of (23), then this type of energy is indeed submodular. Consequently (38) can be solved with α-expansions. □

Figure 2 provides an illustration of the notation for edge weights in a simplified situation. In order to solve problem (38) with the α-expansion algorithm, we propose to define the capacities c of edges $\mathcal{E}$ in the graph $\mathcal{G}_\alpha$ for all $\{u, v\}$ pairs of neighbours, as described in Table 1.

5.3 Other Methods

Other combinatorial optimization methods might also be used. For instance, when minimizing *isotropic* TV as in (22), one might want to use Chambolle's algorithm [37].

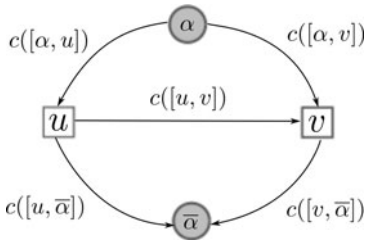


Fig. 2 Notations for the α-expansion graph, following Kolmogorov *et al.* [36]. Here we took a simplified 2-pixel neighbourhood. The cost (or capacity) between u and v is labelled as $c([u,v])$ for instance, and so on for all edges. The expressions for the capacity for all edges are given in Table 1

Table 1 Capacities for the α-expansion graph of Fig. 2

Edge	Capacity[a]
$c([u,\overline{\alpha}])$	$\mathcal{R}(K_u)+\sum_{(u,v)\in\mathcal{N}}\mathcal{R}(A_{u,v}-C_{u,v})+\sum_{(v,u)\in\mathcal{N}}C_{v,u}$
$c([\alpha,u])$	$\mathcal{R}(-K_u)+\sum_{(u,v)\in\mathcal{N}}\mathcal{R}(C_{u,v}-A_{u,v})$
$c([u,v])$	$\sum_{(u,v)\in\mathcal{N}}(B_{u,v}+C_{u,v}-A_{u,v})$

[a]The following notation is used: $\mathcal{R}$ denotes the ramp function, i.e. $\mathcal{R}(x)=0$ if $x\in(-\infty,0)$ and $\mathcal{R}(x)=x$ if $x\in[0,+\infty)$, $K_u=\varphi_{n_u,m_u}(r_{i_{\mathcal{P}}(n_u,m_u)},f(n_u,m_u))-\varphi_{n_u,m_u}(r_\alpha,f(n_u,m_u))$, $A_{u,v}=\psi(|i_{\mathcal{P}}(n_u,m_u)-i_{\mathcal{P}}(n_v,m_v)|)$, $B_{u,v}=\psi(|i_{\mathcal{P}}(n_u,m_u)-\alpha|)$, $C_{u,v}=\psi(|\alpha-i_{\mathcal{P}}(n_v,m_v)|)$

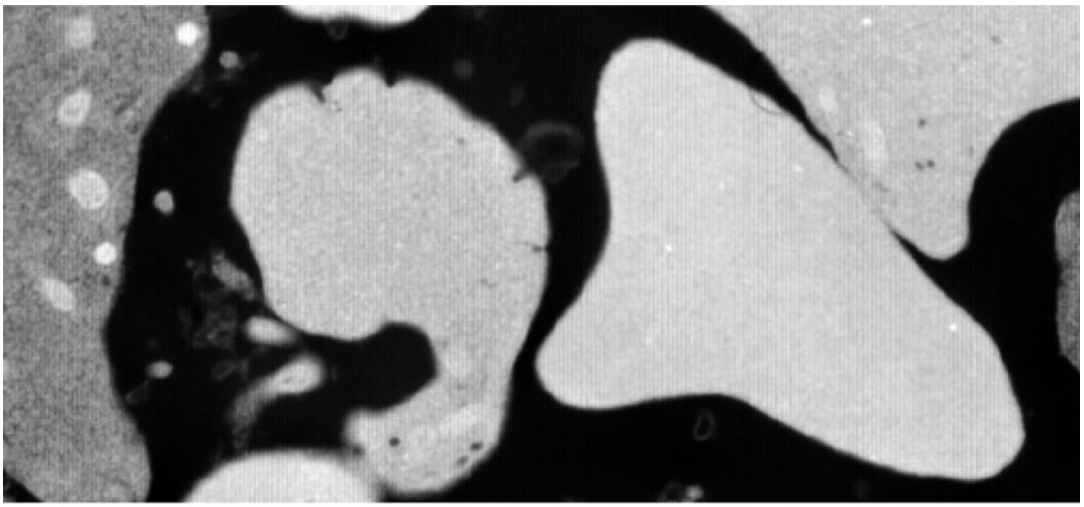
Similarly to the Ishikawa framework, we would obtain the global optimum in this case also. Moreover *isotropic* TV minimization was recently discussed among others by Lellmann *et al.* [38], Trobin *et al.* [39] and Zach *et al.* [40]. One other possibility is the use of α–β generalized range moves algorithm, which is shown in [41] to be able to optimize a wider range of combinatorial energies than α-expansion method presented in Sect. 5.2. Furthermore, similar properties are held by the FastPD [42] and the PD3$_a$ [43] algorithms, both introduced by Komodakis *et al.* Also worth mentioning is the Darbon and Sigelle method for levelable energies, introduced in [44], the Kolmogorov and Shioura primal and primal-dual algorithms and Zalesky's MSFM algorithm [45], since they are all faster than Ishikawa's approach, while still providing an exact solution for a similar class of functions. Our method can be also improved using higher order cliques, which already has been proven to provide effective filtering results [46]. It might be also possible to extend the quantization techniques proposed by Chambolle and Darbon in [47]. Also of interest would be to explore variants of anisotropic diffusion, and other combinatorial optimizers such as generalized Dirichlet solvers, which are naturally multi-label [48] and could provide much simpler algorithms.

6 Simulation Examples

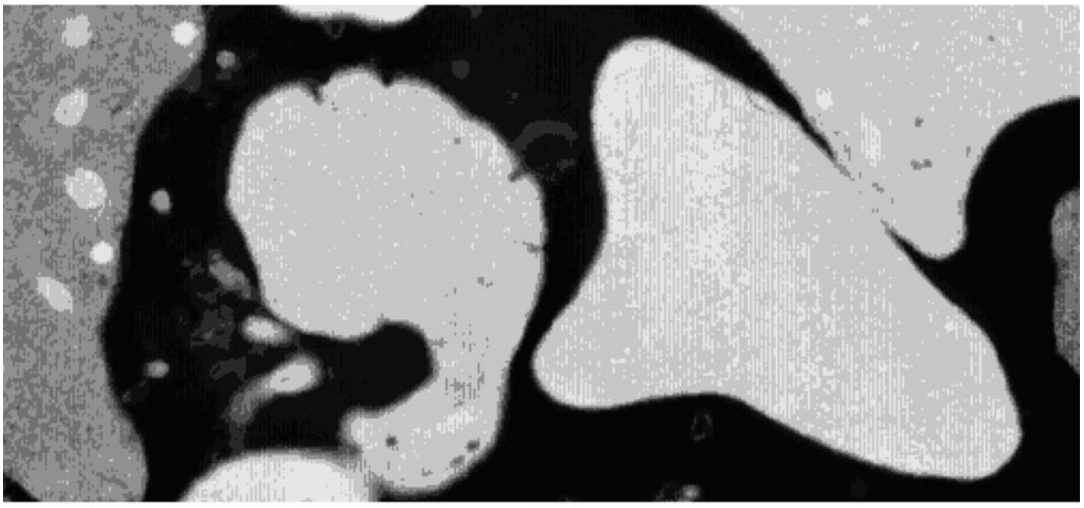
In this section we present four experiments in order to demonstrate the performance of our method in various scenarii. Both color and grey scale images are considered. For grey scale images, our approach is confronted with the LM method [3]. It is a fair comparison, since the same function Φ is used for both algorithms. Although sophisticated initialization procedures [49–51] can be employed for LM and our approach, the methods presented in the following were simply initialized with either uniform or cumulative histograms based decision levels. In the case of color images, we compared our method with: (i) special case of LBG algorithm with Φ defined as ℓ_2 norm (K-means), (ii) median cut [52] and (iii) Wu's method [53]. Ximagic (http://www.ximagic.com) quantization package was used to generate results of K-means, Wu and median cut algorithms. Their performance is measured in terms of SNR between the original and quantized images and also by the Shannon entropy of order $(2,2)$ (that is the entropy over image blocks of size 3×3). Note that, in all the following experiments, regularization functions are used corresponding to a 4-pixel neighbourhood (2 pixels in horizontal and 2 in vertical direction) in the employed graph cut techniques. They were implemented with the help of the publicly available library described in [54]. When running experiments using Algorithm 3, there are 4 parameters to set. We have set $\omega_1=\omega_2=\omega_3=1/3$ and λ_ℓ was fixed and equaled 1.5. The appropriate choice of parameter μ depends on the ratio between maximum values of Φ and ρ codomain, the level of noise in original image and prior knowledge about the desired entropy of output images.

6.1 Low Resolution Quantization

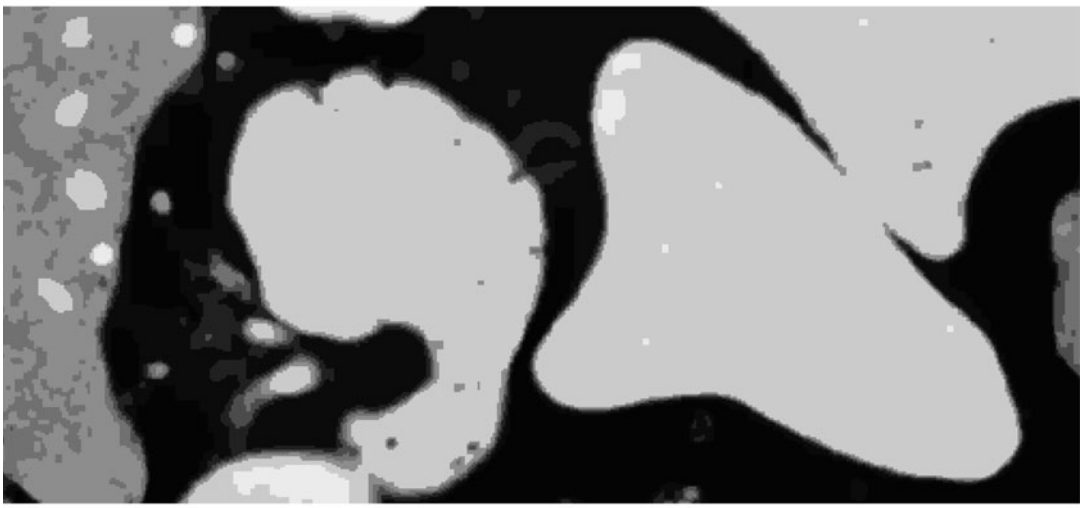
First, we consider grey scale image quantization over $Q=8$ levels. The combinatorial method described in Sect. 5.1 was used to find the global optimum of convex criterion (18) with function Φ defined as the ℓ_1 norm and function ρ defined by (23) where ψ is the identity. It was applied to 8 bit microscopy image of size 512×512 (from public domain, http://www.remf.dartmouth.edu), the fragment of which is shown in Fig. 3(a). Regularization parameter μ was hand-optimized to 10. Both methods, LM and ours, were initialized with uniform decision levels. In order to solve (36), Algorithm 3 was used. The convex set C is defined by (29), where $\delta=12$. As expected, our results provide the best spatial smoothness among the considered methods, which is confirmed by the entropy equal to 0.58 bpp, while in case of LM it is equal to 0.84 bpp. In this example, it is shown that, in case of quantization with high level reduction, our method provides smaller entropy rate while maintaining the desired fidelity.



(a)



(b)



(c)

Fig. 3 Figures (**a**, **b**, **c**) illustrate a fragment of the original image, LM and our results, respectively. Note that LM retained acquisition vertical artifacts, which are absent in our result

In the second example, we show that a similar behaviour is obtained for different choices of Φ, regularity criterion and combinatorial method. This time, the number of quantization levels is $Q = 32$, function (18) is specified by Φ defined as the squared ℓ_2 norm and ρ defined by (23) where ψ is the binary cost-function (25). It is applied to the color-image of size 256×256, which is shown in Fig. 4(a). Figure 4(e) presents the results when μ is set to 25 and in Fig. 4(f), when it is set to 50. The difference between the two presented images (Fig. 4(e) and Fig. 4(f)) is not significant but highlights the visual influence of parameter μ. The criterion (18) was minimized by using the modified α-expansion graph described in Sect. 5.2, which was initialized with $\boldsymbol{r}^{(0)}$ obtained by median cut algorithm. Image pixels were mapped into the XYZ image space [55]. Similarly to the previous example, Fig. 4 shows that a better spatial smoothness is obtained with the proposed approach. This is also verified by inspecting the entropy value, which in our case is equal to 1.06 bpp for $\mu = 25$, and 1.00 bpp for $\mu = 50$, whereas in the case of Wu, K-means and Median-cut the entropies are equal to 1.18 bpp, 1.14 bpp, 1.19 bpp, respectively.

6.2 Quantization in the Presence of Noise

Next, we present the performance of our method in the presence of noise. Note that here function ϕ is chosen based on two noise models, i.e. ℓ_2 for Gaussian and ℓ_1 for Laplacian noise. Firstly, the problem of grey scale image quantization over 16 levels is investigated. The image of size 256×256, shown in Fig. 5(b), is corrupted by zero-mean i.i.d. Laplacian noise with standard deviation 9. Quantization is performed using Algorithm 2. The method described in Sect. 5.2 is used to minimize energy (18), where Φ is defined as the ℓ_1 norm and ρ is given by (23) with ψ taken as the truncated linear function (24), where the limiting constant is set to $\zeta = 3$. The associated regularization parameter μ was experimentally chosen equal to 6. Both methods, LM and ours, were initialized with cumulative histogram based decision levels. Problem (36) was solved by using Algorithm 3. The convex set C is defined by (29), where $\delta = 1$. The proposed approach shows satisfactory results when dealing with Laplacian noise: (i) the visual effect of the noise is reduced (see Fig. 5(d)), (ii) the SNR, which was equal to 22.7 dB for the noisy image increases to 24.6 dB, and (iii) the entropy is only 0.96 bpp. In case of LM (see Fig. 5(c)), the SNR is equal 22.4 dB and the entropy is 1.41 bpp. In this example, we show that, in case of quantization in the presence of noise, our method reconstructs the original image, while performing image quantization.

Similar properties have been observed for $D > 1$. To illustrate this fact, the quantization over 16 quantization levels of a 300×300 color image is presented in Fig. 6. Zero-mean Gaussian noise with standard deviation 20 was added to the image presented in Fig. 6(a) (source: photo by Neon JA, colored by Richard Bartz / Wikimedia Commons). This image was transformed from the RGB space into a more appropriate one, using the linear transformation defined by the matrix of its PCA (Principal Component Analysis) components. Then, the total order of quantization levels along the principal component is chosen, which corresponds to $\eta^\top = [1\,0\,0]$. The convex set C is defined by (29). Since the probability of merging codevectors is negligible in three-channel color space, the associated parameter δ was set to 0. The resulting image (see Fig. 6(f)) was obtained by minimizing energy (18), which was initialized with decision levels computed by the median cut method. Function Φ was defined by (6) and ρ by (23) where ψ is identity and μ is equal to 250. The algorithm described in Sect. 5.1 was used for computing $i_{\mathcal{P}}^{(\ell)}$. One can observe that the noise has been highly reduced in our result (Fig. 6(f)), while the K-means method (Fig. 6(d)) preserved noise in the images. This is also verified by SNR values which is equal to 13.8 dB for our method and 10.6 dB, 10.4 dB and 9.8 dB for the K-means, Wu and median-cut, respectively. The difference is even greater in terms of entropy: our method led to 0.79 bpp and the other ones to 1.48 bpp. Additionally, the quantization result for the original image is presented. Our result (Fig. 6(e)) was obtained with the same algorithm settings as described above except μ, which here is equal to 30 and of course the PCA parameters, which were computed from the original image. Our method performs the required quantization and provides an interesting tradeoff between precision and smoothness, which is validated with an SNR of 18.5 dB and an entropy of 0.9 bpp. In contrast, K-means (Fig. 6(c)) achieved a SNR = 20.2 dB and an entropy = 1.1 bpp.

6.3 Note about Computation Time

The time complexity of Algorithm 2 is equal to the product of the complexity of each iteration and the complexity of the number of iterations ℓ. The bound on ℓ is not known a priori. Our observation suggests that it is a function of the weight of smoothness term μ, number of quantization levels Q, and the spatial entropy of original image f. Moreover, there may be small differences in the number of iterations, depending on the choice of the combinatorial optimization method. For instance, the first problem described in Sect. 6.1, which was solved with an Ishikawa-like graph, converges in 18 iterations. In contrast, using the α-expansion algorithm (Algorithm 4), it converges in only 16. In practice the number of iterations never seems to exceed 50 for grey-scale and 200 for color images. By analyzing the inner loop of Algorithm 2, one can observe that the complexity of step 1 is greater than the one of step 2. Thus, the computation time of each iteration is strongly dominated by the cost of step 1, namely finding $i_{\mathcal{P}}^{(\ell)}$. Note that Algorithm 3 is run only if matrix $\boldsymbol{r}$ derived from a centroid rule does not belong to C, so usually its

Fig. 4 (**a**) is the original image, (**b**, **c**, **d**) the Wu, K-means and median-cut results respectively; (**e**, **f**) show our result for μ equal 25 and 50 respectively. Note that there are many isolated small regions in (**b**, **c**, **d**), while both (**e**) and (**f**) feature only smooth large regions, retaining global aspect nonetheless

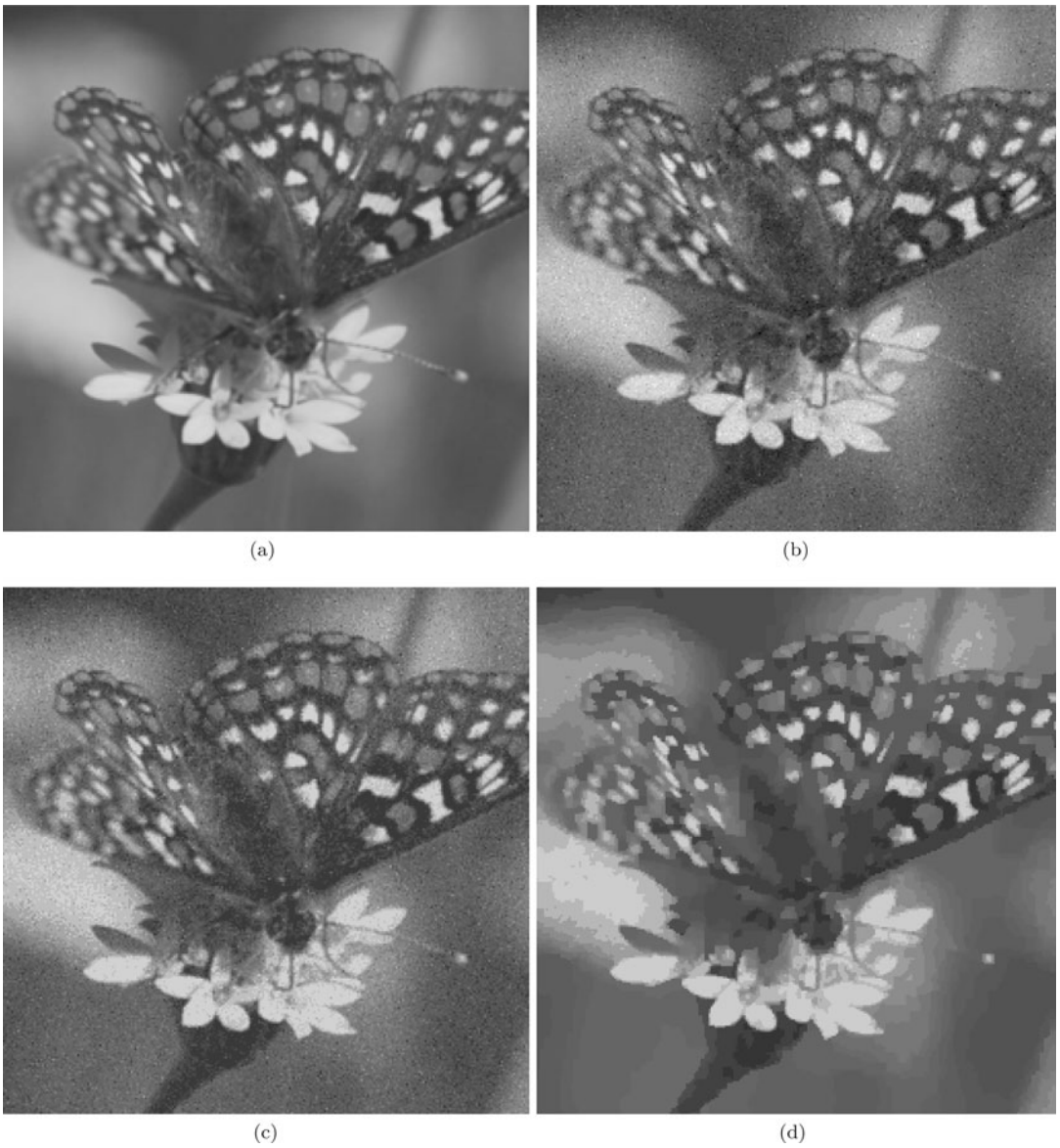


Fig. 5 (**a**) original image, (**b**) noisy version, and (**c**, **d**) LM and our result, respectively

influence on the overall time complexity of Algorithm 2 is small for grey-scale images. It becomes more important for multi-channel images. Generally, the cost of combinatorial graph-cut based methods depends on $|\mathbb{V}|$ and on the number of quantization levels Q. More precisely, the relabeling algorithm finds solution for single graphs in polynomial time $O(|\mathcal{V}|^3)$, where $|\mathcal{V}|$ is equal to $Q \times |\mathbb{V}|$ for the method described in Sect. 5.1 and to $|\mathbb{V}|$ for the method described in Sect. 5.2. However, Algorithm 4 (in Sect. 5.2) requires solving many different graphs independently, so its computation cost increases linearly with the number of quantization levels Q. It is worth noting that some recently published extensions of the α-expansion algorithm are faster. In particular, Lempitsky *et al.* presented the LogCut and Fusion move methods that lead to nearly logarithmic growth [56], e.g. for $Q = 256$ the algorithm converges approximately 10 times faster. A similar acceleration was obtained by the FastPD algorithm introduced by Komodakis *et al.* [42] and analyzed by Kolmogorov in [57]. Likewise, recently introduced primal and primal-dual algorithms by Kolmogorov *et al.* in [58] may be an alternative for the method described in Sect. 5.1. They offer a significant improvement in terms of time efficiency. For instance, our first problem described in Sect. 6.1 solved with the method described in Sect. 5.1 takes 37 seconds, while using Kolmogorov's primal only algorithm, it takes only 12 seconds. As an alternative to the methods presented in Sect. 5, one may adopt these novel methods. Constant progress in the efficiency of graph-cut algorithms makes our approach increasingly competitive with the ones that do not feature a smoothness constraint. Nonetheless,

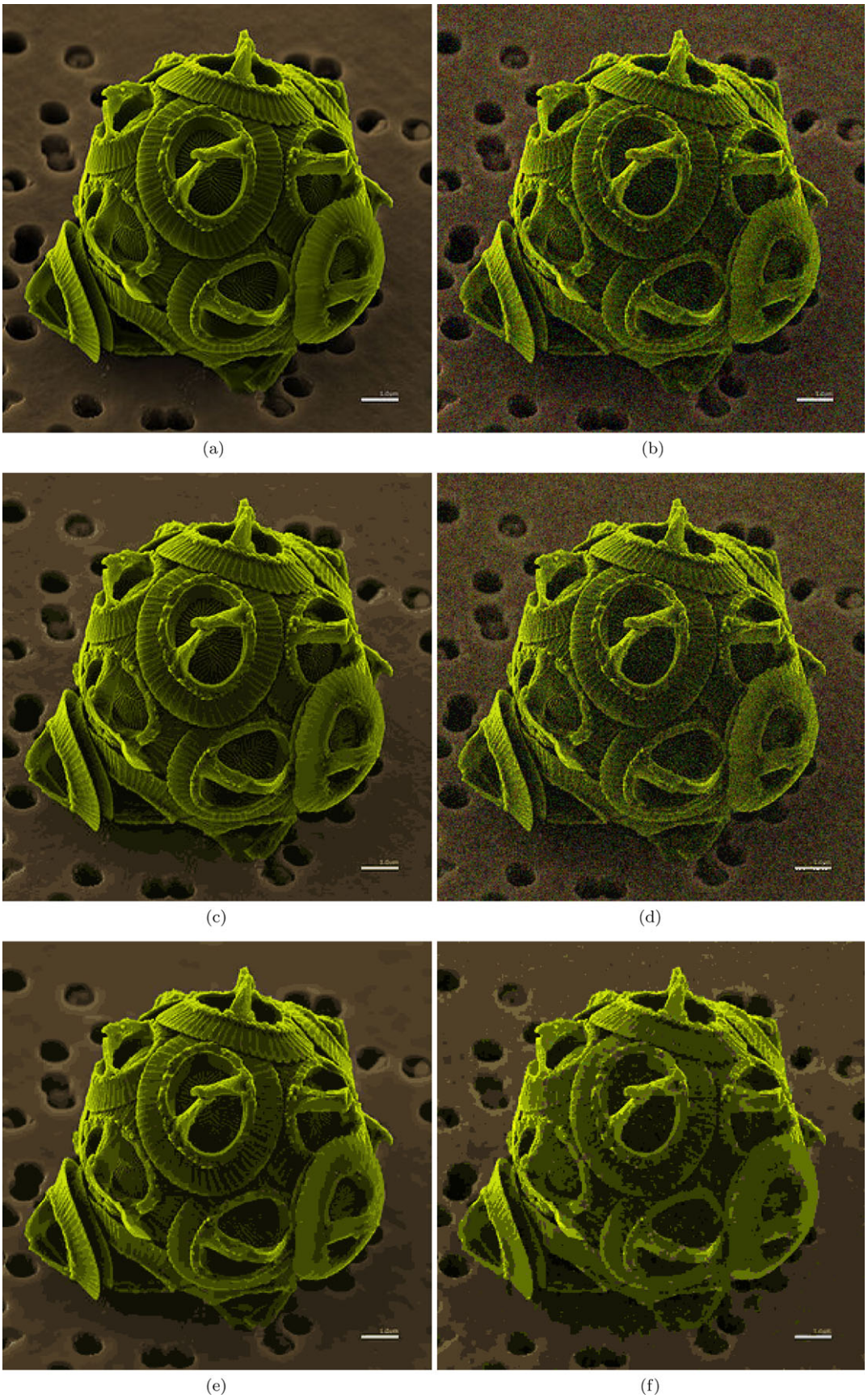


Fig. 6 (**a**) original image, (**b**) its noisy version, (**c**, **e**) and (**d**, **f**) K-means and our result for clear and noisy case, respectively

Table 2 Iteration number and computation time for all examples presented in the paper. The values in the brackets for Ex2 concerns case with $\mu = 50$, and without brackets with $\mu = 25$. The values in and without brackets for Ex4 concerns case without and with noise, respectively

	Ex1	Ex2	Ex3	Ex4
No. of iter.	18	183 (113)	6	51 (42)
Time [s]	37	1618 (1015)	27	808 (625)

our method may take significantly more time than the use of basic quantization methods (for details see Table 2). The tests were performed single-threaded, on a computer with a 2.5 GHz Intel Xeon processor, in the RedHat Enterprise Linux 5.5 environment, using the GCC compiler version 4.1 in 64-bit mode.

7 Conclusion

In this paper, we have proposed a new quantization method based on a two-step procedure intertwining a convex optimization algorithm for quantization level selection and a combinatorial regularization procedure. Unlike classical methods, the proposed approach allows us to enforce a tunable spatial regularity in the quantized image. We have also shown that both grey scale and color images can be processed. As shown by our simulation results, the proposed approach leads to promising results, in particular in the presence of noise. As future work, we plan to explore isotropic regularization methods, to adapt and implement faster combinatorial algorithms and to take advantage of this method in various applications such as image compression and multispectral/hyperspectral imaging.

References

1. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic, Dordrecht (1992)
2. Max, J.: Quantizing for minimum distortion. IRE Trans. Inf. Theory **6**(1), 7–12 (1960)
3. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
4. Linde, Y., Buzo, A., Gray, R.: An algorithm for vector quantizer design. IEEE Trans. Commun. **28**(1), 84–95 (1980)
5. Wu, X.: On convergence of Lloyd's method I. IEEE Trans. Inf. Theory **38**(1), 171–174 (1992)
6. Du, Q., Emelianenko, M., Ju, L.: Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. SIAM J. Numer. Anal. **44**(1), 102–119 (2006)
7. Arya, V., Mittal, A., Joshi, R.C.: An efficient coding method for teleconferencing and medical image sequences. In: International Conference on Intelligent Sensing and Information Processing, pp. 8–13 (2005)
8. Kawada, R., Koike, A., Nakajima, Y.: Prefilter control scheme for low bitrate tv distribution. In: IEEE International Conference on Multimedia and Expo, pp. 769–772 (2006)
9. Chuang, K., Tzeng, H., Chen, S., Wu, J., Chen, T.: Fuzzy C-means clustering with spatial information for image segmentation. Comput. Med. Imaging Graph. **30**, 9–15 (2006)
10. Alvarez, L., Esclarín, J.: Image quantization using reaction-diffusion equations. SIAM J. Appl. Math. **57**(1), 153–175 (1997)
11. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
12. Bruce, J.D.: Optimum quantization. Technical Report 429, Massachusetts Institute of Technology, Research Laboratory of Electronics, Cambridge, MA (1965)
13. Wu, X., Rokne, J.: An $O(KN \log N)$ algorithm for optimum k-level quantization on histograms of n points. In: ACM Annual Computer Science Conference, Louisville, KY, pp. 339–343 (1989)
14. Wu, X.: Optimal quantization by matrix searching. J. Algorithms **12**(4), 663–673 (1991)
15. Rockafellar, R.T., Wets, R.J.-B.: Variational Analysis. Springer, Oxford (2004)
16. Veksler, O.: Efficient graph-based energy minimization methods in computer vision. Ph.D. thesis, Cornell University, Ithaca, NY, USA (1999)
17. Vertan, C., Popescu, V., Buzuloiu, V.: Morphological like operators for color images. In: Proc. Eur. Sig. and Image Proc. Conference, Trieste, Italy, 10–13 September (1996)
18. Talbot, H., Evans, C., Jones, R.: Complete ordering and multivariate mathematical morphology. In: ISMM '98: Proceedings of the Fourth International Symposium on Mathematical Morphology and Its Applications to Image and Signal Processing, pp. 27–34. Kluwer Academic, Norwell (1998)
19. Chanussot, J., Lambert, P.: Bit mixing paradigm for multivalued morphological filters. In: Sixth International Conference on Image Processing and Its Applications, vol. 2, Dublin, pp. 804–808 (1997)
20. Hill, B., Roger, Th., Vorhagen, F.W.: Comparative analysis of the quantization of color spaces on the basis of the CIELAB color-difference formula. ACM Trans. Graph. **16**(2), 109–154 (1997)
21. Eckart, C., Young, G.: The approximation of one by another of lower rank. Psychometrika **1**(3), 211–218 (1936)
22. Combettes, P.L., Pesquet, J.-C.: Proximal splitting methods in signal processing. In: Bauschke, H.H., Burachik, R., Combettes, P.L., Elser, V., Luke, D.R., Wolkowicz, H. (eds.) Fixed-Point Algorithms for Inverse Problems in Science and Engineering. Springer, New York (2010)
23. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms. Grundlehren, vol. 305, p. 306. Springer, Berlin (1993)
24. Pesquet, J.-C.: A parallel inertial proximal optimization method. Preprint (2010). http://www.optimization-online.org/DB_HTML/2010/11/2825.html
25. Combettes, P.L., Pesquet, J.-C.: A proximal decomposition method for solving convex variational inverse problems. Inverse Probl. **24**(6) (2008)
26. Setzer, S., Steidl, G., Teuber, T.: Deblurring Poissonian images by split Bregman techniques. J. Vis. Commun. Image Represent. **21**(3), 193–199 (2010)
27. Goldstein, T., Osher, S.: The split Bregman method for L1-regularized problems. SIAM J. Imaging Sci. **2**(2), 323–343 (2009)
28. Afonso, M.V., Bioucas-Dias, J.M., Figueiredo, M.A.T.: Fast image recovery using variable splitting and constrained optimization. IEEE Trans. Image Process. **19**, 2345–2356 (2010)
29. Afonso, M., Bioucas-Dias, J., Figueiredo, M.: A fast algorithm for the constrained formulation of compressive image reconstruction and other linear inverse problems. In: Proc. Int. Conf. Acoust., Speech Signal Process., Dallas, USA (2010)
30. Moreau, J.J.: Proximité et dualité dans un espace hilbertien. Bull. Soc. Math. Fr. **93**, 273–299 (1965)

31. Chaux, C., Combettes, P.L., Pesquet, J.-C., Wajs, V.R.: A variational formulation for frame based inverse problems. Inverse Probl. **23**, 1495–1518 (2007)
32. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
33. Ishikawa, H., Geiger, D.: Mapping image restoration to a graph problem. In: IEEE-EURASIP Workshop Nonlinear Signal Image Process., Antalya, Turkey, 20–23 June 1999. pp. 189–193 (1999)
34. Ford, J.L.R., Fulkerson, D.R.: Flows in Networks. Princeton University Press, Princeton (1962)
35. Ishikawa, H.: Exact optimization for Markov random fields with convex priors. IEEE Trans. Pattern Anal. Mach. Intell. **25**(10), 1333–1336 (2003)
36. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 147–159 (2004)
37. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20**(1–2), 89–97 (2004)
38. Lellmann, J., Kappes, J., Yuan, J., Becker, F., Schnörr, C.: Convex multi-class image labeling by simplex-constrained total variation. In: SSVM '09: Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision, pp. 150–162. Springer, Berlin (2009)
39. Trobin, W., Pock, T., Cremers, D., Bischof, H.: Continuous energy minimization via repeated binary fusion. In: ECCV, Part IV, Marseille, France, 12–18 October 2008, pp. 677–690 (2008)
40. Frahm, M., Zach, J.M., Niethammer, C.: Continuous maximal flows and Wulff shapes: Application to MRFs. In: IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, pp. 1911–1918 (2009)
41. Veksler, O.: Graph cut based optimization for MRFs with truncated convex priors. In: IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, pp. 1–8 (2007)
42. Komodakis, N., Tziritas, G., Paragios, N.: Performance vs computational efficiency for optimizing single and dynamic MRFs: Setting the state of the art with primal-dual strategies. Comput. Vis. Image Underst. **112**(1), 14–29 (2008)
43. Komodakis, N., Tziritas, G.: Approximate labeling via graph cuts based on linear programming. IEEE Trans. Pattern Anal. Mach. Intell. **29**(8), 1436–1453 (2007)
44. Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation, part II: levelable functions, convex priors and non-convex cases. J. Math. Imaging Vis. **26**(3), 277–291 (2006)
45. Zalesky, B.: Efficient determination of Gibbs estimators with submodular energy functions. http://arxiv.org/abs/math/0304041 (2003)
46. Ishikawa, H.: Higher-order clique reduction in binary graph cut. In: IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, pp. 2993–3000 (2009)
47. Chambolle, A., Darbon, J.: On total variation minimization and surface evolution using parametric maximum flows. Int. J. Comput. Vis. **84**(3), 288–307 (2009)
48. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In: Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan, pp. 731–738 (2009)
49. Wu, X.: On initialization of Max's algorithm for optimum quantization. IEEE Trans. Commun. **38**(10), 1653–1656 (1990)
50. Peric, Z., Nikolic, J.: An effective method for initialization of Lloyd-Max's algorithm of optimal scalar quantization for Laplacian source. Informatica **18**(2), 279–288 (2007)
51. Katsavounidis, I., Kuo, C.-C.J., Zhang, Z.: A new initialization technique for generalized Lloyd iteration. IEEE Signal Process. Lett. **1**(10), 144–146 (1994)
52. Heckbert, P.: Color image quantization for frame buffer display. Comput. Graph. **16**(3), 297–307 (1982)
53. Wu, X.: Color quantization by dynamic programming and principal analysis. ACM Trans. Graph. **11**(4), 348–372 (1992)
54. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. **26**, 359–374 (2001)
55. Ohno, Y.: CIE fundamentals for color measurements. In: IS&T NIP16 Conference, Vancouver, Canada, 16–20 October (2000)
56. Lempitsky, V.S., Rother, C., Roth, S., Blake, A.: Fusion moves for Markov random field optimization. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1392–1405 (2010)
57. Kolmogorov, V.: A note on the primal-dual method for semi-metric labeling problem. Technical report, UCL (2007)
58. Kolmogorov, V., Shioura, A.: New algorithms for convex cost tension problem with application to computer vision. Discrete Optim. **6**(4), 378–393 (2009)

Appendix G

Continuous Maximum Flows

This article was published in IEEE TPAMI as Appleton and Talbot (2006).

```
@article{AppletonTalbot2006,
Author = {Appleton, B. and Talbot, H.},
Journal = {IEEE Transaction on Pattern Analysis and Machine Intelligence},
Number = {1},
Pages = {106-118},
Title = {Globally Minimal Surfaces by Continuous Maximal Flows},
Volume = {28},
Year = {2006}}
```

Globally Minimal Surfaces by Continuous Maximal Flows

Ben Appleton and Hugues Talbot

Abstract—In this paper, we address the computation of globally minimal curves and surfaces for image segmentation and stereo reconstruction. We present a solution, simulating a continuous maximal flow by a novel system of partial differential equations. Existing methods are either grid-biased (graph-based methods) or suboptimal (active contours and surfaces). The solution simulates the flow of an ideal fluid with isotropic velocity constraints. Velocity constraints are defined by a metric derived from image data. An auxiliary potential function is introduced to create a system of partial differential equations. It is proven that the algorithm produces a globally maximal continuous flow at convergence, and that the globally minimal surface may be obtained trivially from the auxiliary potential. The bias of minimal surface methods toward small objects is also addressed. An efficient implementation is given for the flow simulation. The globally minimal surface algorithm is applied to segmentation in 2D and 3D as well as to stereo matching. Results in 2D agree with an existing minimal contour algorithm for planar images. Results in 3D segmentation and stereo matching demonstrate that the new algorithm is robust and free from grid bias.

Index Terms—Partial differential equations, graph-theoretic methods, edge and feature detection.

1 Introduction

Geometric optimization methods provide an exciting approach to solving image analysis problems. They have been applied with great success to image segmentation and to stereo reconstruction. They explicitly acknowledge the uncertainty commonly present in the extraction of geometric structures from images due to noise, occlusions, and background clutter and can, in some cases, obtain provably best estimates according to a measure of quality appropriate to the application.

Broadly speaking, there are two classes of geometric optimization techniques. One class is the active contour methods, including snakes [1], level sets [2], [3], and geodesic active contours and surfaces [4], [5]. Another class of methods taking a very different approach is the graph-based methods including shortest paths [6] and graph cuts [7].

Active contour methods model the evolution of a curve or surface toward a structure of interest in an image. They are usually based on a variational approach, performing a gradient descent flow to locally minimize an energy function whose minima ideally correspond to the objects of interest in the image. Unfortunately, the energy functions used in such models typically possess large numbers of local minima due to noise and irrelevant objects and, as a result, active contours are known to be highly dependent on their initialization. A wide array of heuristics have been proposed to assist in avoiding or overcoming these irrelevant minima, including pressure forces designed to overcome shallow minima [8], multiresolution approaches designed to focus on objects which persist at high scales, and methods which modify the gradient descent to favor more significant contours [9]. Despite the advent of these heuristics, active contours typically require manual intervention which limits their application.

Graph-based methods are well-known in image analysis and in stereo matching. Lloyd [10] and Ohta and Kanade [11] were among the first to propose stereo matching by shortest paths. Shortest paths remain competitive in current stereo research as they form the core of a number of minimal surface methods [12], [13]. Graph cuts have also been applied to 3D reconstruction, sacrificing speed for improved accuracy [14]. These methods are also used in image segmentation. Bamford and Lovell [15] segmented cell nucleii using a polar trellis centered on the nucleus. They computed shortest paths using a Viterbi or dynamic programming approach. Graph-based methods may obtain optimal solutions to the associated minimization problem. However, their use is restricted in practice because they suffer from discretization artifacts. These typically result in a preference for contours and surfaces to travel along the grid directions. See [16] for a good introduction to these methods.

Ideally, geometric optimization methods used in image analysis should be free of these problems, being both isotropic and optimal. In recent years, several advances have been made in the extension of optimal methods from discrete graphs to continuous spaces. Dijkstra's classic shortest path algorithm [6] was extended in [17] and [18] to compute minimal geodesics and continuous distance functions. These have found broad application to optimal control, wave propagation, and computer vision. The problem of continuous graph cuts has also received some attention. Hu [19] described a method for approximating continuous minimal surfaces by a cut in a vertex weighted graph. Boykov and Kolmogorov [20] recently proposed a method

- *B. Appleton is with Google, Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043. E-mail: appleton@google.com.*
- *H. Talbot is with IGM-A2SI-ESIEE, BP 99-2 Bd Blaise-Pascal, 93162 Noisy-le-Grand Cedex, France. E-mail: talboth@esiee.fr.*

Manuscript received 30 Sept. 2004; revised 29 Mar. 2005; accepted 26 Apr. 2005; published online 11 Nov. 2005.
Recommended for acceptance by G. Sapiro.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0518-0904.

for computing edge weights which approximate continuous graph cuts, toward the goal of computing globally minimal surfaces for segmentation and stereo vision.

In this paper, we present an algorithm to compute globally minimal curves, surfaces, and partitionings in arbitrary Riemannian spaces. Section 2 introduces Geodesic Active Contours and Surfaces. Section 3 introduces discrete weighted graphs and continuous Riemannian spaces along with a number of relationships between geometric optimization problems. Section 4 then presents an algorithm for obtaining continuous maximal flows in arbitrary Riemannian spaces with scalar metric. Also presented is a proof of correctness and an efficient implementation. Section 5 presents a solution to the inherent bias of minimal surfaces toward small objects. Section 6 presents the results of the application of this new algorithm and Section 7 concludes.

2 Geodesic Active Contours and Surfaces

Caselles et al. introduced Geodesic Active Contours [4] and Geodesic Active Surfaces [5] for segmentation in 2D and 3D images. They are closed curves or surfaces which evolve to minimize their weighted length or area:

$$E[S] = \oint_S g(S)\mathrm{d}S. \tag{1}$$

$E[S]$ is often termed the *energy* of the surface S. In segmentation, $g \geq \varepsilon > 0$ is a soft edge indicator function, tending toward zero where local image features suggest the presence of an object boundary. Caselles et al. [4] also proposed the following form for the metric

$$g = \frac{1}{1 + |\nabla G_\sigma \star I|^p} + \varepsilon. \tag{2}$$

$|\nabla G_\sigma \star I|$ is the magnitude of the gradient at scale σ. It is usually raised to a power $p = 1$ or 2. ε is an arc length or surface area penalty which effectively regularizes the minimal surface. They also demonstrated that all local minima are smooth surfaces for $\varepsilon > 0$.

Geodesic Active Contours and Surfaces form an initial surface via a gradient descent flow toward a local minima of the energy functional. We may derive the gradient descent flow by variational calculus, giving the Euler-Lagrange equation:

$$\frac{\partial S}{\partial \tau} = -\Big(g\kappa - \nabla g \cdot \vec{N}\Big)\vec{N}. \tag{3}$$

Here, τ is the evolution parameter or *time*, $\vec{N} = \frac{\nabla\phi}{|\nabla\phi|}$ is the surface normal, and $\kappa = \nabla \cdot \vec{N}$ is the mean curvature.

The evolution of this surface may be implemented using a level set embedding due to [2]. For a function $\phi : I\!R^N \rightarrow I\!R$ whose zero level set is $S = \{\mathbf{x}|\phi(\mathbf{x}) = 0\}$, we may evolve ϕ so as to implement the gradient descent flow for S given in (3):

$$\frac{\partial \phi}{\partial \tau} = \nabla \cdot \left(g\frac{\nabla\phi}{|\nabla\phi|}\right)|\nabla\phi|.$$

A more efficient, implicit update scheme has also been presented in [21]. Unfortunately, as we pointed out earlier, these gradient descent flows usually converge to local minima with no guarantee on the optimality of the resulting segmentation.

3 Weighted Graphs and Riemannian Spaces

A number of optimal methods have been proposed for computer vision based on discrete graphs [15], [14] and, later, continuous Riemannian spaces [22], [23]. Here, we review the basic theory and definitions of these closely related frameworks.

3.1 Minimal Paths and Geodesics

A graph G is a pair (V, E) consisting of a vertex set V and an edge set $E \subseteq V \times V$. Vertices may be interpreted as points, while edges are lines connecting pairs of points. A weighted graph includes vertex costs $C_V : V \rightarrow I\!R$ and edge costs $C_E : E \rightarrow I\!R$. In this paper, we consider only positive cost functions.

A simple path P is defined as a sequence of unique vertices, while a cycle has equal endpoints so as to form a loop. The length L of a path P is the sum of vertex and edge costs along the path

$$L[P] = \sum_{v \in P} C_V(v) + \sum_{e \in P} C_E(e).$$

The length of a cycle is defined analogously.

A path between two points s and t is a minimal or shortest path if there exists no connected path of lower length. Such paths may be computed using Dijkstra's shortest path algorithm [6], which first computes the distance of each vertex from s before backtracking from t to s.

A Riemannian space R is the continuous equivalent of a weighted graph. It consists of an N-manifold Ω and an associated metric $g : \Omega \rightarrow I\!R$. Here, we consider only positive scalar metrics $g \in I\!R^+$. A simple curve in a Riemannian space is a 1-manifold embedded in Ω which does not pass through itself. A curve C with parameter ξ in the range $[a, b]$ has length

$$L[C] = \int_a^b g(C(\xi))\left|\frac{\partial C}{\partial \xi}\right|\mathrm{d}\xi.$$

A simple curve between two points s and t is a minimal geodesic if there exists no such curve of lower length. Minimal geodesics may be computed using the Fast Marching Method [18], which first computes a distance function from s by wavefront propagation before backtracking by gradient descent from t to s.

3.2 Minimal Cuts and Minimal Surfaces

A partitioning of a graph G decomposes its vertex set into a collection $\Gamma_G = \{V_1, V_2, \ldots\}$ of disjoint subsets:

$$\bigcup_{V_i \in \Gamma_G} V_i = V, \quad V_i \cap V_j = \emptyset \quad \text{for} \quad i \neq j.$$

To each partition Γ_G, we associate a cost $C(\Gamma_G)$, which is the total cost of the edges whose endpoints lie in different partitions:

$$C(\Gamma_G) = \sum_{e \in E^*} C_E(e).$$

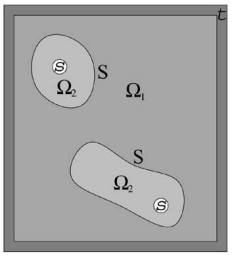


Fig. 1. A binary partitioning of the space Ω. Note that the seeds and the resulting partitions are not necessarily connected.

Here, the *cut* $E^* \subseteq E$ denotes the set of edges crossing the partition. The s-t minimal cut problem seeks the partitioning of minimal cost such that the disjoint vertex sets $s, t \subseteq V$ lie in different partitions. A good introduction to algorithms solving this problem is [16].

A partitioning of a Riemannian space R decomposes the space into a collection $\Gamma_R = \{\Omega_1, \Omega_2, \ldots\}$ of compact subsets whose pairwise intersection has zero Lebesgue measure:

$$\bigcup_{\Omega_i \in \Gamma_R} \Omega_i = \Omega, \quad \mathcal{L}(\Omega_i \cap \Omega_j) = 0 \quad \text{for} \quad i \neq j.$$

Similarly to the discrete case, to each partition Γ_R, we associate a cost $C(\Gamma_R)$, which is the integral of the metric g over the partition surfaces $\partial\Omega_i$,

$$C(\Gamma_R) = \frac{1}{2} \sum_{\Omega_i \in \Gamma_R} \oint_{\partial\Omega_i} g \mathrm{d}(\partial\Omega_i).$$

The potentially confusing term $d(\partial\Omega_i)$ denotes an infinitesimal component of the partition surface $\partial\Omega_i$. Fig. 1 depicts a binary partitioning of the plane Ω. In this paper, we will only consider binary partitionings.

In this continuous case, the s-t minimal cut problem seeks the partition Γ_R of minimal total cost such that the point sets $s, t \subseteq \Omega$ fall in different partitions. To the authors' best knowledge, this paper is the first to solve this problem in continuous spaces with more than two dimensions.

3.3 Maximal Flows

3.3.1 Discrete Case

Let G be a graph with edge costs C_E now reinterpreted as *capacities*. A flow $F_G : E \rightarrow \mathbb{R}$ from a *source* set $s \subseteq V$ to a *sink* set $t \subseteq V$ has the following properties:

- Conservation of flow: The total (signed) flow in and out of any vertex is zero.
- Capacity constraint: The flow along any edge is less than or equal to its capacity:

$$\forall e \in E, \quad F(e) \leq C_E(e).$$

An edge along which the flow is equal to the capacity is described as *saturated*. Ford and Fulkerson [7] demonstrated that the maximal s-t flow equals the minimal s-t cut, with the flow saturated uniformly on the cut. Fig. 2 gives an example of a capacitated graph and an s-t maximal flow through this graph. In this example, s and t are single vertices.

Sedgewick [16] describes how to convert the problem of computing a maximum flow between the sets s and t to an equivalent problem of computing a maximum flow between single vertices s' and t'. First, we add to the graph G two new vertices s' and t', which become the new source and sink, respectively. Then, from s' to each source vertex in s, we add an edge of infinite capacity and, from each sink vertex in t to t', we do likewise. A maximum flow from s' to t' directly corresponds to a maximum flow from s to t in the original graph G. In this paper, we will make implicit use of this direct correspondence between the two viewpoints.

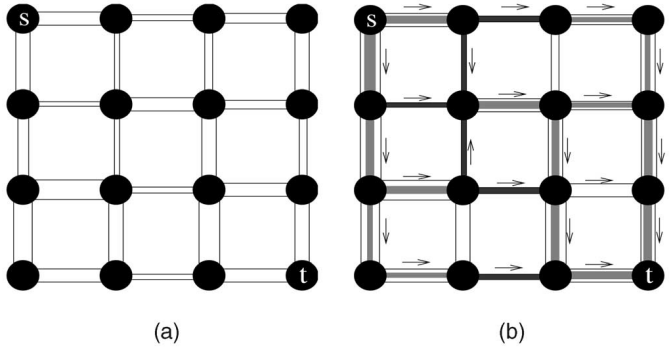


Fig. 2. An example of the minimal cut—maximal flow duality. (a) A capacitated graph. Edge thickness corresponds to capacity. (b) An s-t maximum flow. The set of saturated edges form a minimal cut.

A second convenience which we adopt in this paper is to add an implicit edge connecting $t \rightarrow s$ (equivalently, $t' \rightarrow s'$) with infinite capacity. This ensures that the flow is conserved at every vertex in the graph, rather than treating the source and sink vertices as special cases. With this viewpoint, a maximal flow in a capacitated graph G then maximizes the flow through the $t \rightarrow s$ edge. We denote this flow by F_{st} and its maximization is the objective of the maximal flow problem.

3.3.2 Continuous Case

Strang [24] and Iri [25] explored the theoretical extension of maximal flows to continuous domains. A continuous flow $\vec{F}$ is a vector field over a continuous domain. It has the following properties:

- Conservation of flow: $\nabla \cdot \vec{F} = 0$.
- Capacity constraint: $\left|\vec{F}\right| \leq g$.

In the continous case, the source s and the sink t become compact subsets of the continuous domain.

Let $\vec{F}$ be any flow and S be any simple, closed, and smooth surface containing the source s. Let $\vec{N}_S$ denote the normal to the surface S. The net flow out of the source is denoted F_{st} as in the discrete case. Then, combining the two properties stated above, we obtain:

$$F_{st} = \oint_S \vec{F} \cdot \vec{N}_S \mathrm{d}S \leq \oint_S g \mathrm{d}S. \tag{4}$$

Therefore, all flows are bounded from above by all smooth, simple, and closed surfaces separating the source and sink, and all simple closed surfaces have weighted area bounded from below by all flows from source to sink. In fact, Iri [25] showed that, under very general continuity assumptions, the maximal flow F_{max} is strictly equal to the minimal

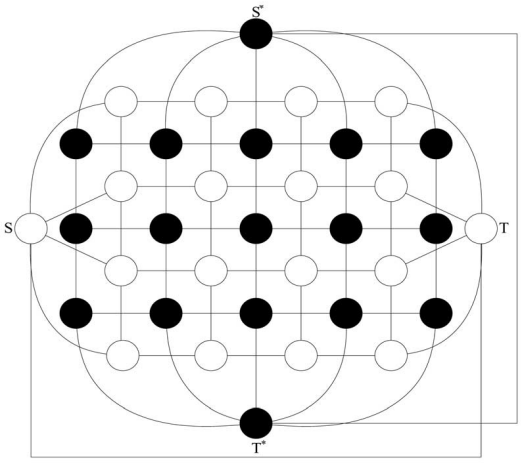


Fig. 3. A planar graph (white) and its dual (black). The vertices of the dual graph correspond to the faces of the primal graph. Edges connect adjacent faces in the dual graph and correspond uniquely to edges in the primal graph.

surface S_{min}. For such a flow and surface, the flow saturates the surface uniformly:

$$\forall \mathbf{x} \in S_{min}, \quad F_{max}(\mathbf{x}) = g(\mathbf{x})\vec{N}(\mathbf{x}). \tag{5}$$

The minimal surface algorithm presented in this paper makes explicit use of this duality.

The duality between maximal flows and minimal cuts and surfaces has a simple interpretation. Any cut forms a bottleneck for a flow, limiting the flow to be less than the capacity of that cut. The maximal flow is limited by all possible cuts and, therefore, must be less than or equal to the cost of the minimal cut. These dualities state that the maximal flow is indeed equal to the minimal cut and, therefore, that a maximal flow saturates a minimal cut.

3.4 The Planar Case

For planar graphs and spaces, some special equivalences exist between, on the one hand, minimal paths and geodesics, and, on the other hand, minimal cuts and surfaces. In the discrete case of a graph embedded in the plane, a minimal cut in this *primal* graph is identical to a shortest path in the *dual* graph whose vertices correspond to the faces of the primal graph. Fig. 3 presents an example of this planar duality between paths and cuts. A similar duality has been noted in the continuous case between geodesics which are manifolds of dimension 1 and minimal surfaces which are manifolds of codimension 1 (and, hence, also dimension 1).

These dualities are important in the design of planar minimal cut algorithms because the computation of shortest paths is more efficient compared to general maximal flow methods. They are used in Weihe's discrete maximal flow algorithm [26] and in Mitchell's continuous maximal flow algorithm [27].

The authors have previously presented an algorithm for 2D image segmentation, Globally Optimal Geodesic Active Contours (GOGAC), which can be interpreted under the planar duality as a solution to the minimal surface problem in 2D spaces. However, despite this connection, the GOGAC algorithm is fundamentally based on the computation of geodesics and cannot be extended beyond two dimensions. The method presented in this paper is based on flows and may therefore be applied to higher dimensional spaces.

3.5 Approximating Minimal Surfaces by Graph Cuts

A number of approaches have been proposed to compute approximate minimal surfaces by transforming the problem to a graph cut. These approaches obtain a polyhedral surface of minimal weighted area, where the weighting is derived from the metric of the original Riemannian space.

Hu [19] presented a formulation of the minimal cut problem in a graph with vertex capacities rather than edge capacities. Under this alternate formulation, a cut becomes a set of vertices whose removal disconnects the source and sink. The cost of a cut is the sum of the capacities of these vertices. The continuous problem is modeled as a grid of square vertices of side length h. The vertex capacities are sampled directly from the metric of the continuous domain. All vertices are connected within a radius $r \gg h$. It was shown that, in the limit as $h \to 0$, $r \to 0$, and $\frac{h}{r} \to 0$, the minimal cut converges to a surface of minimal weighted area. As presented, this method only approximates isotropic metrics.

In Boykov and Kolmogorov [20], an approximation to the minimal surface problem using a graph with edge capacities derived from the metric of the continuous domain. Their approach is able to handle all convex metrics. Edge capacities are derived from the metric of the continuous domain using the Cauchy-Crofton formula from integral geometry.

In both of these approximations, the theoretical convergence of a minimal cut to a minimal surface depends upon the degree of each vertex increasing toward infinity. In practice, the number of directions that each segment of the polyhedral approximation can take on is proportional to the degree of each vertex. For an angular precision of $\Delta\theta$, the degree of each vertex is proportional to $(\frac{1}{\Delta\theta})^N$ in [19] and $(\frac{1}{\Delta\theta})^{N-1}$ in [20]. Consequently, the time and memory required by these algorithms grows rapidly with the desired angular resolution, particularly in higher dimensions.

4 Minimal Surfaces in Three or More Dimensions

In this section, we present a nonlinear system of partial differential equations (PDEs) to compute continuous maximal flows and, hence, obtain globally minimal surfaces. This extends the previous presentation by the same authors in [28], giving a detailed description of the implementation on regular grids and deriving the necessary and sufficient stability conditions.

The development of the following system of PDEs was motivated by considering existing discrete maximum flow algorithms. Two of the more popular maximum flow algorithms are the augmenting-path algorithm of Ford and Fulkerson and the preflow push algorithm of Goldberg and Tarjan [16]. The augmenting-path algorithm maintains a conservative flow at each step, repeatedly searching for paths along which the flow may be increased. However, the direct extension of this algorithm to continuous spaces seems problematic. Not the least of these problems would be the requirement for a nonlocal system, corresponding to

augmenting the flow along curves. This implies that it would not be possible to obtain a partial differential equation framework. Primarily, this is due to the conservation constraint which imposes infinite "stiffness" in the flow, making it difficult to modify the flow locally.

On the other hand, the preflow push algorithm relaxes the conservation constraint, allowing more flow into a vertex than out of it. This algorithm introduces an additional variable at each vertex which, in some sense, ensures that the system converges toward an incompressible flow. This is the approach we take in developing a solution to the continuous maximal flow problem. We allow the flow to have nonzero divergence during its evolution, but introduce a scalar potential field which stores this excess flow. The potential field is then used to drive the flow to become incompressible at convergence.

4.1 A Continuous Maximal Flow Algorithm

The continuous maximal flow system developed in [28] is described by the following system:

$$\frac{\partial P}{\partial \tau} = -\nabla \cdot \vec{F}, \tag{6}$$

$$\frac{\partial \vec{F}}{\partial \tau} = -\nabla P, \tag{7}$$

subject to

$$\left|\vec{F}\right| \leq g. \tag{8}$$

$P = P(\mathbf{x}, \tau) : (\Omega, I\!R^+) \rightarrow I\!R$ is a scalar potential field over the domain Ω evolving over time τ. $\vec{F} = \vec{F}(\mathbf{x}, \tau) : (\Omega, I\!R^+) \rightarrow I\!R^N$ is the vector flow field, also over the N-dimensional domain Ω and evolving over time τ. For boundary conditions, we fix the scalar field P at the source s and sink t: $P(\mathbf{x}) = 1$ for $\mathbf{x} \in s$ and $P(x) = 0$ for $x \in t$. These values are chosen arbitrarily and without loss of generality. Initial conditions may be chosen as $P = 0$ except at the source and sink, and $\vec{F} = 0$ everywhere. However, suitably selected initial conditions may lead to faster convergence, as we will discuss further in Section 4.4.

Equation (6) relaxes the conservation constraint, instead storing excess flow in the potential field P. (7) couples the flow $\vec{F}$ to the potential P such that gradients in the potential drive the flow. (6) and (7) form a simple system of wave equations. They may be viewed as a linear model of the dynamics of an idealized fluid with pressure P and velocity $\vec{F}$, ignoring convection terms. (8) constitutes a hard constraint on the magnitude of the flow velocity $\vec{F}$.

4.2 Properties of the Continuous Maximal Flow Algorithm

4.2.1 Conservation of Potential P

Let $P_A = \int_A P \mathrm{d}A$ denote the total integral of P in a given region A not including s, t. Then, for smooth P and $\vec{F}$,

$$\frac{\partial P_A}{\partial \tau} = -\oint_{\partial A} \vec{F} \cdot \vec{N}_{\partial A} \mathrm{d}(\partial A). \tag{9}$$

So, P is conserved in the interior of any *sourceless* region A (any region not including the source s or sink t).

4.2.2 Monotonic Reduction of Energy $\frac{1}{2}(P^2 + \|\vec{F}\|^2)$

Consider the temporal rate of change of the total quantity of $\frac{1}{2}(P^2 + \|\vec{F}\|^2)$ in a given region A not including s, t. For smooth P and $\vec{F}$,

$$\frac{\partial}{\partial \tau} \int_A \frac{1}{2} \left(P^2 + \|\vec{F}\|^2\right) \mathrm{d}A = -\oint_{\partial A} P \vec{F} \cdot \vec{N}_{\partial A} \mathrm{d}(\partial A). \tag{10}$$

Note that we have momentarily ignored the magnitude constraint (8). Consequently, $\frac{1}{2}(P^2 + \|\vec{F}\|^2)$ is conserved in the interior of any sourceless region A. Including the magnitude constraint may only decrease $\|\vec{F}\|^2$ and, hence, the energy $\frac{1}{2}(P^2 + \|\vec{F}\|^2)$ must monotonically decrease in the interior of a sourceless region. Since the energy is positive, it must converge. To ensure smoothness and convergence of P and $\vec{F}$ independently, a dissipative term can be added to the equations. In the discretised system, this term is not necessary.

4.3 Correctness at Convergence

At convergence, any isosurface of P may be taken as the globally minimal surface S_{min} separating s and t.

Proof. Setting temporal derivatives to zero at convergence, we may restate the system (6), (7), (8):

$$\nabla \cdot \vec{F} = 0$$

$$\begin{array}{llll} \nabla P = 0 & & & \text{if} \left|\vec{F}\right| < g \\ \nabla P = -\lambda \vec{F} & \text{where} & \lambda \geq 0 & \text{if} \left|\vec{F}\right| = g. \end{array}$$

The first equation simply restates the conservation of flow. The second equation is derived from (7), (8). It states that, where $\vec{F}$ is not saturated, P must be constant and, where $\vec{F}$ is saturated, ∇P must be such that $\vec{F}$ cannot change direction or decrease in magnitude. Consequently, $\nabla P \cdot \vec{F} \leq 0$, indicating that P is a (non-strictly) monotonic function along the flow lines of $\vec{F}$. As $\vec{F}$ is divergence-free, flow lines may only initiate at s and terminate at t. Therefore, there are no local extrema in P.

Now, consider the closed region $A_p = \{\mathbf{x} | P(\mathbf{x}) \geq p\}$ obtained from P by the application of a threshold $0 < p < 1$. Due to the monotonicity of P, this is a connected region containing the source s. On the isosurface $S = \partial A_p$, we have $\nabla P \neq \vec{0}$ by construction. Therefore, the flow is uniformly saturated outward on this surface and we obtain:

$$\nabla \cdot \vec{F}_s = \oint_S \vec{F} \cdot N_S \mathrm{d}S = \oint_S g \mathrm{d}S.$$

Hence, $\vec{F}$ and S satisfy (5) for optimality. Therefore, at convergence, any isosurface of P is a globally minimal surface. In the usual case of a unique minimal surface, S_{min} will be the only isosurface at convergence and, hence, P will approach an indicator function for the interior of S_{min}. □

4.4 Implementation

Equations (6) and (7) are discretized on a staggered grid using an explicit first-order scheme in time and space. The

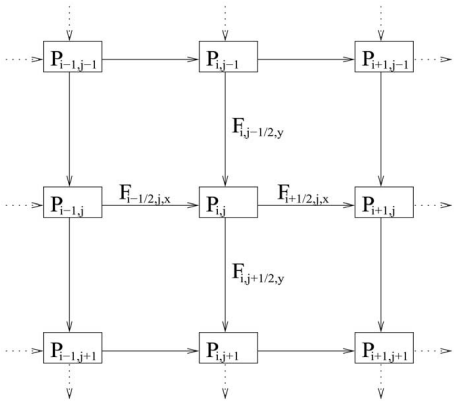


Fig. 4. The discrete representation of the numerical scheme presented here. P is stored at grid vertices while $\vec{F}$ is stored by component on grid edges.

scalar field P is stored on grid points while the vector field $\vec{F}$ is stored by component on grid edges, depicted in Fig. 4. The system of equations is iterated sequentially with the flow magnitude constraint (8) enforced after each timestep.

Here, for simplicity, we describe the update scheme for a single iteration in two dimensions. The spatial grid step is set to $h = 1$. We first consider the linear portion of the update scheme, implementing (6) and (7). We begin by defining the notation that we will use to describe this discrete system. Let n denote the iteration number and $\Delta\tau$ the timestep. Let $P^n_{i,j}$ denote the value of the potential at time $n\Delta\tau$ and gridpoint (i, j) and let $g_{i,j}$ denote the value of the metric at the point (i, j). Let $F^n_{i\pm\frac{1}{2},j,x}$ and $F^n_{i,j\pm\frac{1}{2},y}$ represent the components of the flow along the four edges incident on the point (i, j) at time $n\Delta\tau$. Then, we may give the explicit discretization of the PDE system as:

$$\begin{aligned} &P^{n+1}_{i,j} = P^n_{i,j} - \\ &\Delta\tau\Big(F^n_{i+\frac{1}{2},j,x} - F^n_{i-\frac{1}{2},j,x} + F^n_{i,j+\frac{1}{2},y} - F^n_{i,j-\frac{1}{2},y}\Big), \end{aligned} \tag{11}$$

$$\begin{aligned} F'^{n+1}_{i+\frac{1}{2},j,x} &= F^n_{i+\frac{1}{2},j,x} - \Delta\tau(P^{n+1}_{i+1,j} - P^{n+1}_{i,j}) \\ F'^{n+1}_{i,j+\frac{1}{2},y} &= F^n_{i,j+\frac{1}{2},y} - \Delta\tau(P^{n+1}_{i,j+1} - P^{n+1}_{i,j}). \end{aligned} \tag{12}$$

The magnitude constraint is applied immediately following the update of the flow velocity field by (12). Here, we describe the application of the magnitude constraint at point (i, j) for time $(n + 1)\Delta\tau$, consisting of three stages:

1. Determine the maximal outward flow along each axis:

$$\begin{aligned} \left|F^{n+1}_{i,j,x}\right|' &= \max\Big(-F'^{n+1}_{i-\frac{1}{2},j,x},\ 0,\ F'^{n+1}_{i+\frac{1}{2},j,x}\Big), \\ \left|F^{n+1}_{i,j,y}\right|' &= \max\Big(-F'^{n+1}_{i,j-\frac{1}{2},y},\ 0,\ F'^{n+1}_{i,j+\frac{1}{2},y}\Big). \end{aligned}$$

2. Compare the absolute maximal outward velocity to the metric $g_{i,j}$: If

$$v'^{n+1}_{i,j} \equiv \sqrt{\left(\left|F^{n+1}_{i,j,x}\right|'\right)^2 + \left(\left|F^{n+1}_{i,j,y}\right|'\right)^2} > g_{i,j},$$

then

$$|F^{n+1}_{i,j,x}| = |F^{n+1}_{i,j,x}|' \frac{g_{i,j}}{v'^{n+1}_{i,j}}, \quad |F^{n+1}_{i,j,y}| = |F^{n+1}_{i,j,y}|' \frac{g_{i,j}}{v'^{n+1}_{i,j}}.$$

3. Apply the magnitude constraint to each outward velocity component:

$$\begin{aligned} F^{n+1}_{i-\frac{1}{2},j,x} &= \max\Big(F'^{n+1}_{i-\frac{1}{2},j,x},\ -|F^{n+1}_{i,j,x}|\Big) \\ F^{n+1}_{i+\frac{1}{2},j,x} &= \min\Big(F'^{n+1}_{i+\frac{1}{2},j,x},\ |F^{n+1}_{i,j,x}|\Big) \\ F^{n+1}_{i,j-\frac{1}{2},y} &= \max\Big(F'^{n+1}_{i,j-\frac{1}{2},y},\ -|F^{n+1}_{i,j,y}|\Big) \\ F^{n+1}_{i,j+\frac{1}{2},y} &= \min\Big(F'^{n+1}_{i,j+\frac{1}{2},y},\ |F^{n+1}_{i,j,y}|\Big). \end{aligned}$$

Despite the complexity of its formal description, this update scheme is simple enough that a single implementation is used to handle input data of arbitrary dimension. This explicit scheme is also simple to parallelize by domain decomposition.

Several heuristics have been found to increase the speed of convergence. The fields P and $\vec{F}$ are rapidly initialized using the preflow push discrete maximal flow algorithm with both global and gap relabeling [16]. A multiscale approach is also applied recursively for rapid convergence at a fine grid resolution from a coarse grid initialization. Computation may be avoided in the interior of the source s and sink t, yielding great savings when they occupy a significant portion of the space.

At convergence in the continuous system, in the usual case of a single surface of globally minimal value, the potential field P is theoretically perfectly binary with value 1 within the volume bounded by the minimal surface, and 0 outside. However, in the discrete implementation, convergence is deemed to be attained if the sum of the relative areas of potential $|A_{P\geq 1-\gamma}|$ and $|A_{P\leq\gamma}|$ is greater than μ percent. For example, $\gamma = 0.03$ and $\mu = 99$. Once convergence has been obtained, the minimal surface is extracted from P as the isosurface of value $\frac{1}{2}$ using a bilinear interpolation.

4.5 Stability

In this section, we derive the maximum timestep for which the update scheme described by (11) and (12) is stable. In this analysis, we neglect the magnitude constraint as it may only reduce the magnitudes of the variables of interest and, hence, cause the system to tend toward stability. For simplicity, we perform the derivation in the 2D case and then give the general solution.

By an appropriate combination of (11) and (12), we may obtain the discrete update equation solely for P:

$$\begin{aligned} &P^{n+2}_{i,j} - 2P^{n+1}_{i,j} + P^n_{i,j} = \\ &\Delta\tau^2\Big(-4P^{n+1}_{i,j} + P^{n+1}_{i+1,j} + P^{n+1}_{i-1,j} + P^{n+1}_{i,j+1} + P^{n+1}_{i,j-1}\Big). \end{aligned} \tag{13}$$

This is a discrete analogue to the wave equation $\frac{\partial^2 P}{\partial\tau^2} = \nabla^2 P$, which may be derived from (6) and (7).

Equation (13) describes a linear system and, so, is amenable to spectral analysis. Specifically, consider the

Z-transform over $z_x,\ z_y,\ z_\tau \in \mathbb{C}$ with $|z_x| = |z_y| = 1$ for a bounded field P:

$$P(z_x, z_y, z_\tau) = \sum_{i,j,n} P^n_{i,j} z_x^i z_y^j z_\tau^n. \tag{14}$$

For $P \neq 0$, substitution into (13) gives

$$z_\tau^2 - 2z_\tau + 1 = \Delta\tau^2\left(-4 + z_x + z_x^{-1} + z_y + z_y^{-1}\right). \tag{15}$$

For a stable and causal system, we require $|z_\tau| < 1$. Now, the right side of (15) takes values in the range $[-8(\Delta\tau)^2,\ 0]$ over the entire spatial spectrum. The left side has range $(-4,\ 0]$. Therefore, in order that this equation have a solution for all spatial frequency components, we require that it have a solution when the right side equals $-8(\Delta\tau)^2$, so $\Delta\tau < \frac{1}{\sqrt{2}}$. More generally, when the update is performed in N dimensions, it is simple to show that $\Delta\tau < \frac{1}{\sqrt{N}}$ as before. The same argument may be applied to the evolution of $\vec{F}$ but is not pursued here for reasons of space. This condition on the time step is therefore necessary and sufficient to obtain a stable discrete implementation.

5 Metric Weighting Functions

Minimal surface methods have an inherent bias in favor of small surfaces. In many applications, this is undesirable, resulting in incorrect or even trivial solutions. In this section, we present a technique to automatically remove this bias.

5.1 Construction

Consider a metric that is uniformly constant throughout the domain, $g = 1$. This metric conveys no preference for any particular point through which the partition surface should pass. Intuitively, then, every point in the domain should belong to some (globally) minimal surface. Unfortunately, as the minimal surface problem is posed, this is not the case. In order to improve the behavior of the solutions to this problem, then, we replace the metric g by $g' = gw$, introducing an appropriate weighting function w. This weighting function will account for the geometry of the sources and sinks, so that the minimal surface depends only on the data as represented by g.

Appleton and Talbot [23] considered the special case of a single point source p in a planar image. Here, it was demonstrated that the introduction of the weighting function $w(\mathbf{x}) = \frac{1}{|\mathbf{x}-\mathbf{p}|}$ resulted in a continuum of minimal surfaces, the set of all circles centered on p. In N dimensions, it is simple to see that the modified weighting function $w(\mathbf{x}) = \frac{1}{|\mathbf{x}-\mathbf{p}|^{N-1}}$ will behave similarly, ensuring that each point in the domain belongs to a minimal surface (a hypersphere centered on p). These weighting functions may be extended to other seed geometries. For a line source in three dimensions, we obtain the weighting function $w(x) = \frac{1}{|\mathbf{x}-\mathbf{p}|}$, where $\mathbf{p}$ is the nearest point to $\mathbf{x}$ on the line. More generally, for a set of seeds which form an M-dimensional manifold embedded in N dimensions, we should expect a weighting function that decays as $\frac{1}{|\mathbf{x}-\mathbf{p}|^{N-M-1}}$ in the neighborhood of the manifold.

We wish to derive an *unbiased* flow $\vec{F}$ from which we may define the weighting function $w = \|\vec{F}\|$. This flow will be produced by the source set s and absorbed by the sink set t,

$$\nabla \cdot \vec{F} = \rho, \tag{16}$$

where ρ is a distribution that is zero in the interior of the domain, positive on the source set s and negative on the sink set t, with total source weight $\int_s \rho \mathrm{d}V = 1$ and sink weight $\int_t \rho \mathrm{d}V = -1$. There will naturally be many such flows; here, we select a flow to minimize a measure of the weighting function

$$E[w] = \int_V \frac{1}{2} w^2 \mathrm{d}V = \int_V \frac{1}{2} \|\vec{F}\|^2 \mathrm{d}V.$$

In this way, we will ensure that the weighting function is not arbitrarily large at any particular point in space, as it could be, for example, for some flows with large rotational components.

We may minimize the measure $E[\vec{F}] \equiv E[w]$ by variational calculus: Consider adding a minimization parameter τ to obtain $w \equiv w(\mathbf{x}, \tau)$. Then, we may compute the first variation with respect to τ to determine the local minima of $E[\vec{F}]$:

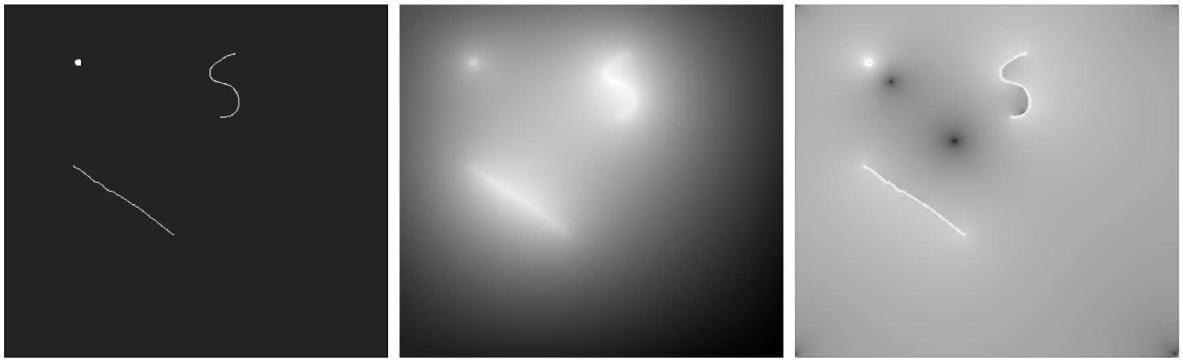
$$\frac{\delta E[\vec{F}]}{\delta \tau} = \int_V \vec{F}_\tau \cdot \vec{F} \mathrm{d}V = 0.$$

Here, we have set the first variation to 0 to obtain a local minimum condition on $E[w]$. This minimization must be carried out subject to the incompressibility constraint expressed in (16). Taking the time derivative of the constraint, we obtain the equivalent constraint $\nabla \cdot \vec{F}_\tau = 0$. Therefore, $\vec{F}_\tau$ may be decomposed into cyclic components, and $\vec{F}$ is a local minimum of $E[\vec{F}]$ if it is locally minimal with respect to all cyclic flows. Consider, then, $\vec{F}_\tau = \vec{T}_C$, the unit tangent vector over the tube formed from the set of all points within a vanishing radius r of the smooth closed curve C, with $\vec{F}_\tau = 0$ elsewhere. For $\vec{F}$ a local minimum of $E[\vec{F}]$, we have

$$\begin{aligned}\frac{\delta E[\vec{F}]}{\delta \tau} &= \int_V \vec{F}_\tau \cdot \vec{F} \mathrm{d}V \\ &= A_{N-1}(r) \oint_C \vec{F}_\tau \cdot \vec{T}_C \mathrm{d}C,\end{aligned}$$

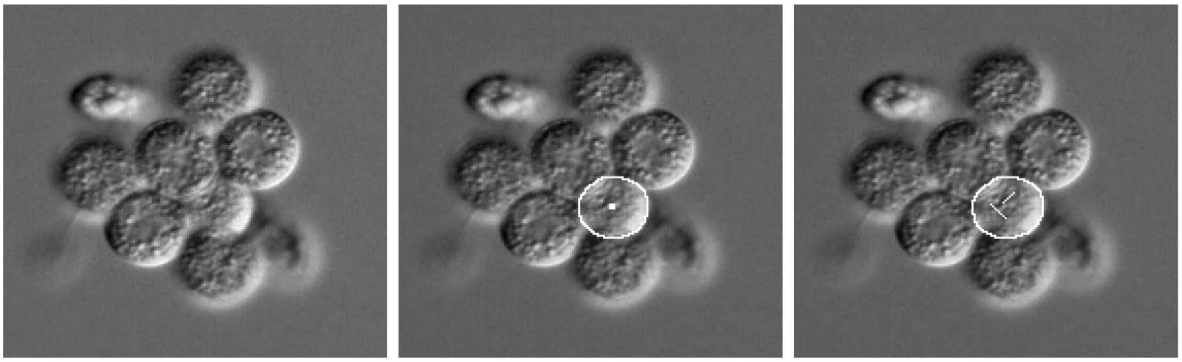
where $A_{N-1}(r)$ is the volume of the $N-1$ dimensional sphere of radius r. So, for $\frac{\delta E[\vec{F}]}{\delta \tau} = 0$, we find that the vector field is a potential flow. Set $\vec{F}_\tau = \nabla\phi_\tau$, then, and replace the divergence of the flow $\vec{F}$ in (16) by the Laplacian of ϕ to obtain $\nabla^2\phi = \rho$. We choose boundary conditions $\lim_{|\mathbf{x}|\to\infty} \nabla\phi(\mathbf{x}) = 0$ so that the flow is zero at infinity. ϕ is then determined up to the addition of a constant which will not affect the weighting function $w = |\nabla\phi|$.

Observe, now, that all isosurfaces S of ϕ in the interior of the domain have constant net flux $\oint_S \nabla\phi \cdot \vec{N}_S \mathrm{d}S = 1$. As $w = |\nabla\phi|$, we then obtain $\oint_S w \mathrm{d}S = 1$ over all isosurfaces of ϕ, with $\oint_S w \mathrm{d}S \geq 1$ for all closed surfaces S containing the seeds. So, the isosurfaces of ϕ form the set of minimal



(a) (b) (c)

Fig. 5. Metric weighting example. (a) The seed geometry. Source points are depicted, while the sink points are the image boundary. (b) The function ϕ computed by convolution. (c) The metric weighting w computed from the numerical derivative of ϕ.



(a) (b) (c)

Fig. 6. An example of the application of metric weighting. (a) A microscope image of a protist, *Chilomonas Paramecium*. (b) Segmentation using a single internal seed. (c) Segmentation with complex seed geometry.

surfaces under the metric w. In general, we have $\nabla\phi \neq 0$ almost everywhere; therefore, almost every point in the domain belongs to some minimal weighted surface under the metric w as desired.

5.2 Implementation

For the regular grids considered in this paper, the weighting functions may be computed by convolving the distribution ρ with the Green's function $\phi_\odot$ with the property $\nabla^2\phi_\odot = \delta(\mathbf{x})$. In $\mathbb{R}^2$, this is $\phi_\odot = \frac{1}{2\pi}\ln(|\mathbf{x}|)$, while, in $\mathbb{R}^3$, this is $\phi_\odot = -\frac{1}{4\pi|\mathbf{x}|}$ [29]. This convolution may be efficiently computed on discrete images using the Fast Fourier Transform. The gradient of ϕ may then be numerically estimated in the discrete grid to obtain the weighting function w.

Fig. 5 shows an example of a set of seed points and the process of computing an appropriate weighting function. The weighting function is highest in the neighborhood of point sources and at the endpoints of line sources.

Fig. 6 depicts the application of metric weighting in the segmentation of a microscope image of a protist, *Chilomonas Paramecium*. Presented are segmentations using a simple seed geometry and a complex seed geometry. The metric weighting scheme proposed in this section produces similar results on the two examples, demonstrating that it does not significantly bias the segmentation.

6 Results

In this section, we demonstrate the results of using globally minimal surfaces for 2D and 3D medical image segmentation and for stereo matching. All results were obtained using the metric weighting scheme introduced in Section 5, except where otherwise noted. Timings were obtained on a quad 2.2GHz AMD Opteron Processor 848 under the Linux operating system. The algorithm presented here has been implemented in C with no assembly optimizations. Timings for minimum cuts have been obtained using the Boost Graph Library implementation of the preflow-push algorithm [30]. The preflow-push algorithm is generally accepted as a fast general-purpose maximum flow algorithm, although a faster image-specific maximum flow algorithm presented in [31] has not been considered here.

6.1 Two-Dimensional Image Segmentation

Object boundaries are often difficult to detect along transitions to adjacent objects with similar features. Segmentation via minimal contours uses the regularization of the segmentation contour to avoid leaking across such gaps. The authors have previously developed an algorithm, Globally Optimal Geodesic Active Contours (GOGAC) [23], which efficiently computes globally minimal contours in planar Riemannian spaces using the planar duality in Section 3.4. Here, we apply discrete minimal cuts, GOGAC, and the algorithm presented in this paper to segment a

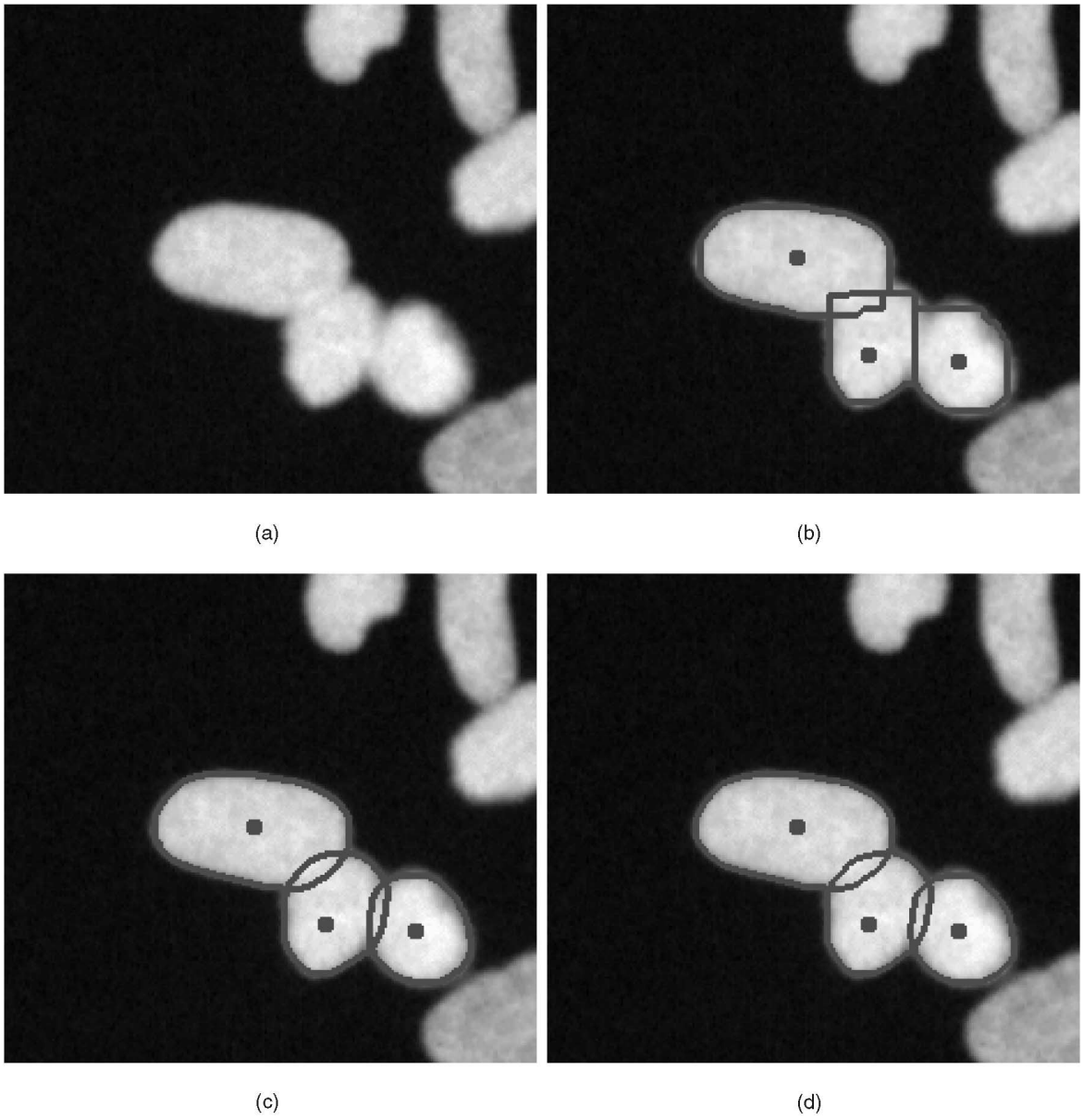


Fig. 7. Segmentation of a cluster of cells in a microscope image. (a) The original image. (b) Discrete minimal cuts. (c) Globally Optimal Geodesic Active Contours. (d) Globally minimal surfaces.

microscope image of a cluster of cells (Fig. 7a) and compare the results. In spite of its apparent simplicity, this problem demonstrates the challenge of delineating faint boundaries between cells without leaking.

We compute a metric (Fig. 7b) from the microscope image as described in (2), with default parameters: $p = 1$, $\varepsilon = 0$, and $\sigma = 1$. Low metric regions are dark, while high metric regions are bright. The regions of low metric correspond to the boundaries of the cells, except where the cells overlap. The metric has been weighted according to the method described in Section 5 (not displayed).

The segmentation of each cell is performed independently in sequence for each method. The source sets are depicted in Figs. 7b, 7c, and 7d, while the sink is the image boundary. The discrete minimal cut produces a clear grid bias and a poor segmentation. GOGAC and the continuous maximal flow algorithm solve the same continuous optimization problem and are in close agreement. Note that the continuous segmentations follow the perceived cell contours despite the weakness of local cues.

The image depicted in Fig. 7a has dimensions 231×221. We reduce the amount of computation required by expanding the sink to include only the cells of interest, a region of size 150×100.

The discrete minimal cuts required 0.41 seconds to compute in total. GOGAC required 0.73 seconds to compute in total. The continuous minimal surface algorithm presented here required 0.68 seconds in total to converge.

6.2 Three-Dimensional Image Segmentation

Here, we demonstrate the application of globally minimal surfaces to a 3D segmentation problem. Fig. 8 depicts a Computed Tomography scan of a chest in which the two lungs are segmented. We compare the results from the application of globally minimal surfaces to those obtained using geodesic active surfaces and discrete

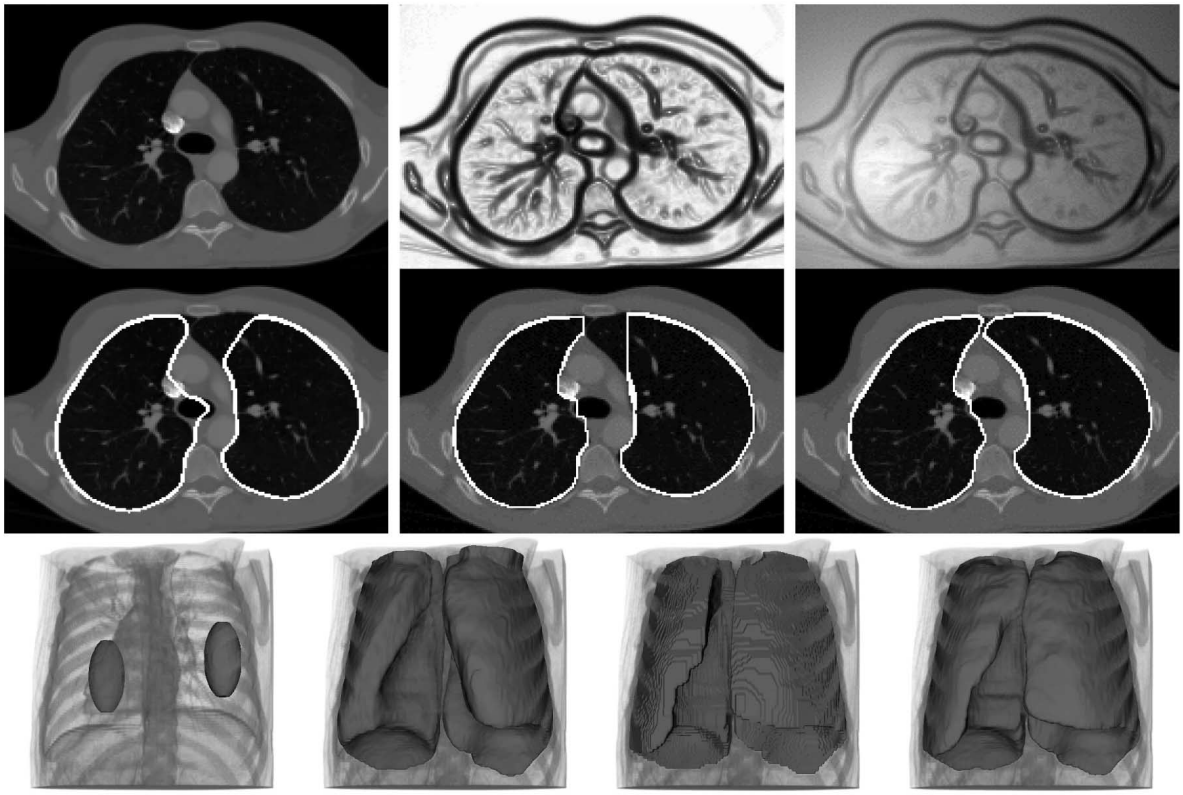


Fig. 8. Segmentation of the lungs in a CT image of a chest. Top row: 2D slices of original data, metric, and weighted metric (log scale). Middle row: 2D slices of segmentations by geodesic active surfaces, minimal cuts, and globally minimal surfaces. Bottom row: 3D views of seeds and segmentations by geodesic active surfaces, minimal cuts, and globally minimal surfaces.

minimal cuts. The minimal cut and minimal surface segmentations use the same input, i.e., the same weighted metric, as well as the seeds. The sources are ellipsoids inside each lung, while the sinks are the five volume boundaries not including the uppermost face where the lung is open. The geodesic active surfaces are unable to use the weighted metric due to the large range of metric values produced. Instead, they use the unweighted metric and are initialized from large ellipsoids inside each lung. An additional artificial inflation term was also used to drive the level sets to fill the lungs. The two lungs are segmented independently in all three methods.

The top row of Fig. 8 depicts corresponding 2D slices of the original CT data, the metric derived from this data, and the weighted metric for the right lung. The weighted metric has been displayed on a logarithmic scale due to its large range of values. The middle row of Fig. 8 shows corresponding 2D slices of segmentations by each of the methods considered here: geodesic active surfaces, minimal cuts, and globally minimal surfaces. The bottom row of Fig. 8 provides 3D views of the sources as well as the segmentations obtained by the different methods.

The geodesic active surfaces produce a poor segmentation. At the top of the lung, the inflation term is too strong, causing the surface to leak through the weak edges of the lung. Elsewhere, the surface has failed to completely fill the lung. This behavior is common in the application of active contour methods and, as in this case, sometimes hard to avoid.

The discrete minimal cuts also produce inaccurate segmentations. Observe the bias toward the grid directions, which can be clearly seen as the flat boundaries in the interior surfaces at the top of the lungs. By contrast, the continuous minimal surface does not exhibit such directional bias, giving a faithful segmentation.

The CT data shown in Fig. 8 has dimensions $200 \times 160 \times 90$. The Geodesic Active Surfaces required 279 seconds to converge to the final result. The discrete minimal cuts required 44 seconds to compute using the preflow push algorithm. The continuous minimal surface algorithm required only 28.8 seconds using three scales. The minimal surface algorithm uses a multiscale framework to obtain a fast initialization from the solution at a coarser scale. The fields P and $\vec{F}$ are initialized using a minimum cut at the coarsest scales. In this example, the minimal surface segmentation is faster than the minimum cut segmentation due to the multiscale framework.

6.3 Three-Dimensional Scene Reconstruction from Stereo Images

The reconstruction of a 3D scene from two or more images is often performed using an energy minimization approach [13], [14]. Here, we adapt the framework of [14], replacing their discrete graph cut by a globally minimal surface.

In stereo matching, a number of metrics have been proposed for real and synthetic images. Here, we use the zero-mean normalized cross correlation (ZNCC) window-based matching score, which performs well on natural scenes with lighting variation and specular reflections and may be computed very efficiently [12]. We set $g = 1 - ZNCC$ to convert high matching scores to low metrics suitable for minimization. Matching scores are computed using a

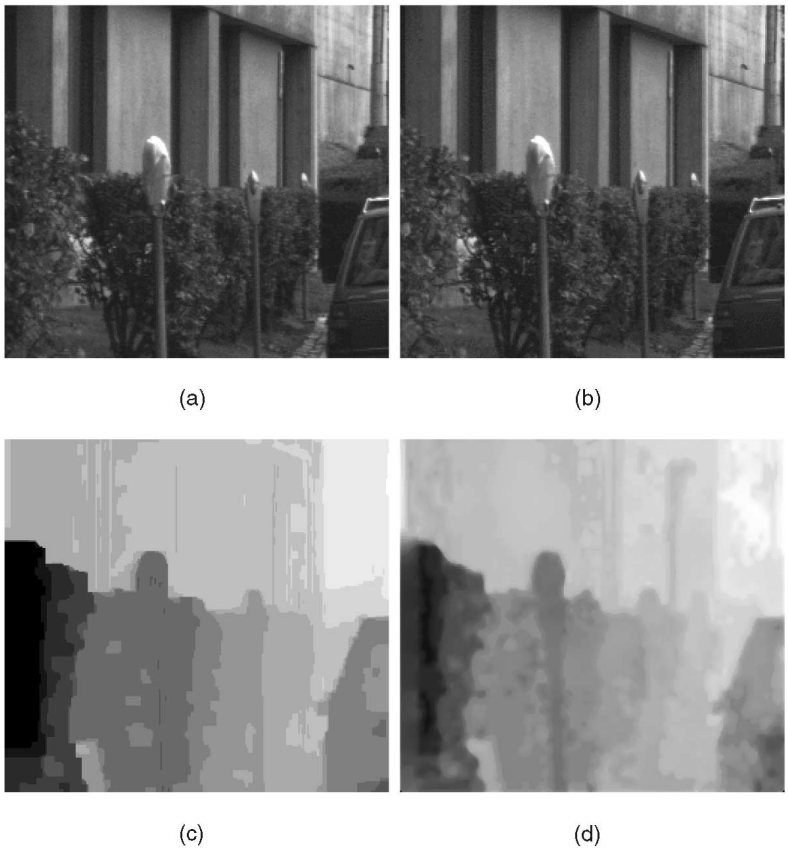


Fig. 9. Stereo matching from two views. (a) and (b) The original images. (c) Disparity map obtained by a discrete maximal flow. (d) Disparity map obtained by a globally minimal surface. Note the improved detail.

5×5 window. The stereo pair being analyzed has a disparity range of $[-15,\ 0]$. Following [14], the source and sink are connected to the first and last layers of the disparity volume. Both the discrete minimal cut and the globally minimal surface are computed from the same metric.

The results of the stereo matching are depicted in Fig. 9. These results are shown as disparity maps (depth maps) as well as surface meshes. We observe that the discrete minimal cut produces large flat regions due to the small number of disparities and, hence, poor depth resolution of discrete methods. Compared to the graph cut, we can see a great deal more detail in the disparity map computed by the globally minimal surface. This includes the surface texture of the bushes as well as the third parking meter. In addition to the improvement in depth resolution is the rotational invariance of the continuous method. This can be seen on the frame of the car, where the discrete method produces a "rectangular" curve, while the minimal surface produces a straight line.

The stereo image pair used here has dimensions 256×240. The discrete minimal cut required 11.3 seconds to compute. The continuous minimal surface algorithm required only 8.3 seconds.

6.4 Accuracy

In the continuous theory, the system of PDEs presented in (6), (7), and (8) was proven to obtain the globally minimal surface at convergence. However, in order to develop a practical algorithm, it was necessary to discretize these equations. An explicit, first-order finite difference discretization was presented in Section 4.4. It is natural to question whether this discretization introduces grid bias into the solution surfaces.

To address this question, we compare the surface obtained by the proposed discretization with the analytic solution on a simple problem in three dimensions, the computation of a catenoid. Consider two circles of equal radius R' whose centers lie along the z-axis. These circles lie in the planes $z = -H$ and $z = H$, respectively. The minimal surface which connects these two circles is a catenoid. An illustration of such a nontrivial minimal surface is sometimes given using soap bubbles. The surface points of this catenoid (x, y, z) are described by:

$$x = R' \cosh\Big(\frac{z}{R'}\Big) \cos(\theta),$$
$$y = R' \cosh\Big(\frac{z}{R'}\Big) \sin(\theta),$$

where $\theta \in [0, 2\pi)$ and $z \in [-H, H]$. Here, R' is selected to meet the boundary condition $R' \cosh\big(\frac{H}{R'}\big) = R$.

We set $R = 35$ and $H = 15$ and compare the analytic and numeric solutions. The algorithm proposed in this paper is discretized on a $120 \times 120 \times 31$ grid with grid step $h = 1$. For the Euclidean metric, we set $g = 1$ everywhere, without metric weighting. For this problem, we may enforce the circular boundary conditions by placing disk-shaped sources of radius R on the vertical boundaries $z = H$ and $z = -H$, and sinks elsewhere on the volume boundary. The results for this comparison are presented in Fig. 10. Fig. 10a depicts the analytic solution. Figs. 10b and 10c depict the

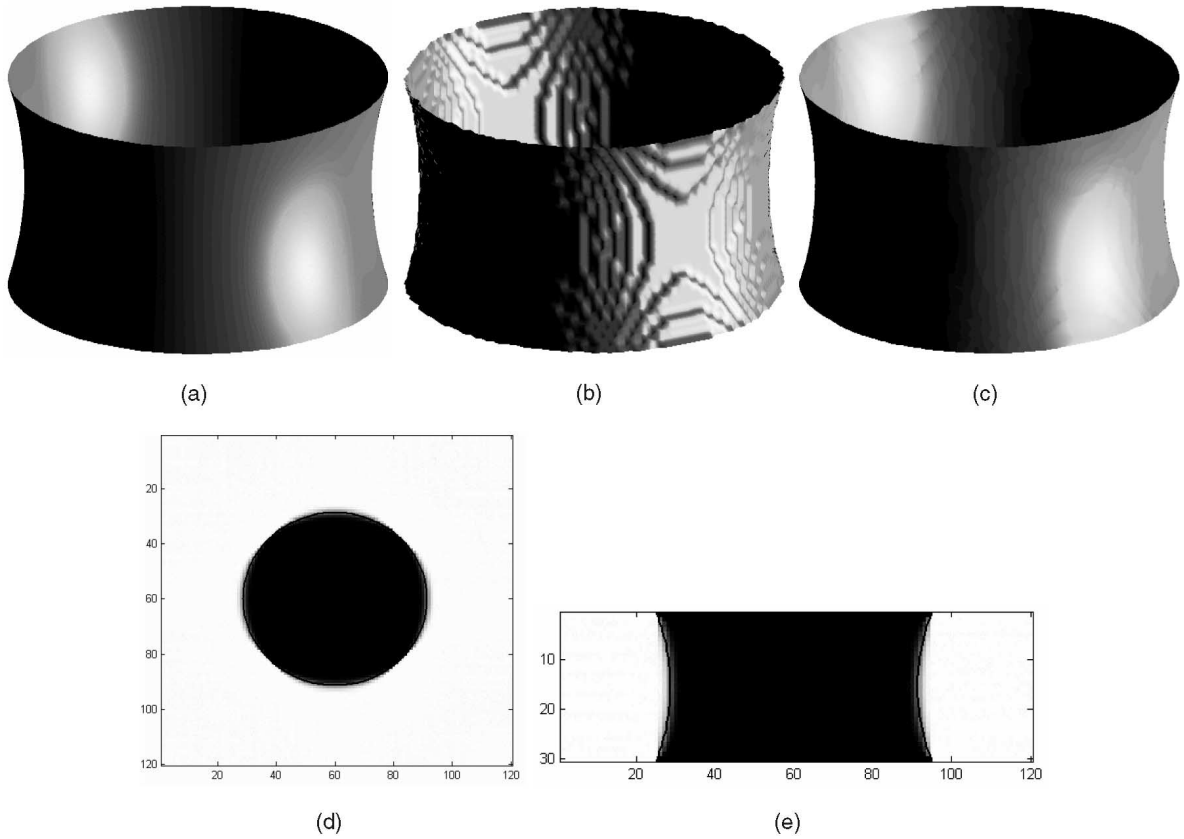


Fig. 10. The catenoid test problem. (a) The correct minimal surface, constructed analytically. (b) The discrete minimal surface obtained by thresholding $P \leq 0.5$. (c) The minimal surface obtained as the isosurface $P = \frac{1}{2}$. (d) A horizontal slice through P. The correct cross-section is overlayed in black. (e) A vertical slice through P. The correct cross-section is overlayed in black.

numeric solution obtained from P, respectively, by thresholding and by isosurface extraction. The isosurface and the analytic solution are in clear agreement. By contrast, the threshold result demonstrates that it is important to extract isosurfaces in order to avoid discretisation artifacts. This result is also the best that a discrete method such as a graph cut could obtain without increasing the grid resolution. Fig. 10d depicts a horizontal slice $z = 0$ through the potential function P computed by the new algorithm, overlayed with the corresponding cross-section of the analytic solution (a circle of radius R'). Fig. 10e depicts a vertical slice $x = 0$ through P, overlayed with the corresponding cross-section of the analytic solution (two catenaries). In both Figs. 10c and 10d, the analytic solution closely coincides with the isosurface $P = \frac{1}{2}$.

In order to make a quantitative comparison, we measured the average distance between the analytic surface and the computed isosurface. The mean distance between these surfaces was 0.09 while the root-mean-square distance was 0.11. In this simple example then, the new algorithm obtains a result which is accurate to approximately 10 percent of the grid step.

7 Conclusions

In this paper, we have developed a new algorithm to compute globally minimal weighted surfaces for image segmentation and stereo matching.

We obtain these surfaces using a nonlinear system of partial differential equations which simulate an ideal fluid flow. The velocity of the flow is constrained in magnitude by a spatially varying metric, itself derived from the image or images being analyzed. This simulation is performed using a simple finite difference scheme with explicit update step. To improve efficiency, a multiresolution scheme is used to reduce computational costs and the solution is approximated at the coarsest scale by a discrete maximal flow.

A proof is given that, at convergence, the algorithm produces a globally maximal flow. The minimal surface is simply an isosurface of the auxiliary potential function. A proof that the system converges is left to future work.

Results are given demonstrating the application of globally minimal surfaces to 2D and 3D segmentation and to stereo matching. Comparison to an existing optimal geodesic active contour for 2D images demonstrates close similarity. Comparisons for 3D segmentation and stereo matching demonstrate that globally minimal surfaces overcome the existing problems with graph-based approaches and with active contours. On a simple test problem with known analytic solution, the discrete implementation of this algorithm was shown to be accurate to 10 percent of the grid step size. The algorithm is also efficient when compared to previous methods. These results suggest that many existing applications using geodesic active surfaces or graph cuts will benefit from the improved accuracy and reliability of globally minimal surfaces.

ACKNOWLEDGMENTS

This work was carried out when B. Appleton was affiliated with the School of ITEE, The Univerity of Queensland, QLD 4067, Australia, and H. Talbot was affiliated with CSIRO, Mathematics and Information Sciences, Sydney, Australia. The authors would like to acknowledge Simon Long of the University of Queensland for an interesting discussion leading to the numerical implementation of the metric weighting scheme presented in this paper. We would also like to thank Rob Dunne of CSIRO Mathematical and Information Sciences for the use of his computer to obtain the results in Section 6.

REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision,* vol. 1, no. 4, pp. 321-331, 1998.
[2] S. Osher and J.A. Sethian, "Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations," *J. Computational Physics,* vol. 79, pp. 12-49, 1988, citeseer.nj.nec.com/osher88fronts.html.
[3] J.A. Sethian, *Level Set Methods and Fast Marching Methods—Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* Cambridge Univ. Press, 1999.
[4] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," *Int'l J. Computer Vision,* vol. 22, no. 1, pp. 61-79, 1997.
[5] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert, "Minimal Surfaces Based Object Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, pp. 394-398, 1997.
[6] E. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik,* vol. 1, pp. 269-271, 1959.
[7] J.L.R. Ford and D.R. Fulkerson, *Flows in Networks.* Princeton, N.J.: Princeton Univ. Press, 1962.
[8] L.D. Cohen, "On Active Contour Models and Balloons," *Computer Vision, Graphics, and Image Processing. Image Understanding,* vol. 53, no. 2, pp. 211-218, 1991, citeseer.nj.nec.com/cohen91active.html.
[9] C. Xu and J.L. Prince, "Snakes, Shapes and Gradient Vector Flow," *IEEE Trans. Image Processing,* vol. 7, no. 3, pp. 359-369, Mar. 1998.
[10] S. Lloyd, "Stereo Matching Using Intra- and Inter-Row Dynamic Programming," *Pattern Recognition Letters,* vol. 4, pp. 273-277, Sept. 1986.
[11] Y. Ohta and T. Kanade, "Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 7, no. 2, pp. 139-154, Mar. 1985.
[12] C. Sun, "Fast Stereo Matching Using Rectangular Subregioning and 3D Maximum-Surface Techniques," *Int'l J. Computer Vision,* vol. 47, nos. 1/2/3, pp. 99-117, May 2002.
[13] C. Leung, B. Appleton, and C. Sun, "Fast Stereo Matching by Iterated Dynamic Programming and Quadtree Subregioning," *Proc. British Machine Vision Conf.,* S.B.A. Hoppe and T. Ellis, eds., vol. 1, pp. 97-106, Sept. 2004.
[14] S. Roy and I.J. Cox, "A Maximum-Flow Formulation of the n-Camera Stereo Correspondence Problem," *Proc. Int'l Conf. Computer Vision,* pp. 492-499, Jan. 1998.
[15] P. Bamford and B. Lovell, "Unsupervised Cell Nucleus Segmentation with Active Contours," *Signal Processing,* special issue on deformable models and techniques for image and signal processing, vol. 71, no. 2, pp. 203-213, 1998.
[16] R. Sedgewick, *Algorithms in C,* third ed. Addison-Wesley, 2002.
[17] J.N. Tsitsiklis, "Efficient Algorithms for Globally Optimal Trajectories," *IEEE Trans. Automatic Control,* vol. 40, no. 9, pp. 1528-1538, Sept. 1995.
[18] J. Sethian, "A Fast Marching Level Set Method for Monotonically Advancing Fronts," *Proc. Nat'l Academy of Sciences,* vol. 93, no. 4, pp. 1591-1595, 1996, citeseer.nj.nec.com/sethian95fast.html.
[19] T.C. Hu, *Integer Programming and Network Flows.* Reading, Mass.: Addison-Wesley, 1969.
[20] Y. Boykov and V. Kolmogorov, "Computing Geodesics and Minimal Surfaces via Graph Cuts," *Proc. Int'l Conf. Computer Vision,* pp. 26-33, Oct. 2003.
[21] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, "Fast Geodesic Active Contours," *IEEE Trans. Image Processing,* vol. 10, no. 10, pp. 1467-1475, 2001.
[22] L.D. Cohen and R. Kimmel, "Global Minimum for Active Contour Models: A Minimal Path Approach," *Intl J. Computer Vision,* vol. 24, no. 1, pp. 57-78, Aug. 1997, citeseer.nj.nec.com/cohen97global.html.
[23] B. Appleton and H. Talbot, "Globally Optimal Geodesic Active Contours," *J. Math. Imaging and Vision,* July 2005.
[24] G. Strang, "Maximal Flow through a Domain," *Math. Programming,* vol. 26, pp. 123-143, 1983.
[25] M. Iri, *Survey of Mathematical Programming.* North-Holland, Amsterdam, 1979.
[26] K. Weihe, "Maximum (s,t)-Flows in Planar Networks in $O(|V|\log|V|)$ Time," *J. Computer and System Sciences,* vol. 55, no. 3, pp. 454-475, Dec. 1997.
[27] J.S. Mitchell, "On Maximum Flows in Polyhedral Domains," *Proc. Fourth Ann. Symp. Computational Geometry,* pp. 341-351, 1988.
[28] B. Appleton and H. Talbot, "Globally Optimal Surfaces by Continuous Maximal Flows," *Digital Image Computing: Techniques and Applications, Proc. VIIth APRS Conf.,* vol. 2, pp. 987-996, Dec. 2003.
[29] G. Strang, *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, 1986.
[30] J. Siek, L.-Q. Lee, and A. Lumsdaine, *The Boost Graph Library: User Guide and Reference Manual.* Addison-Wesley, 2002.
[31] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1124-1137, Sept. 2004.

Ben Appleton completed the PhD degree in the area of image analysis at the University of Queensland (UQ), Australia, in 2005. He received degrees in engineering and science from the University of Queensland in 2001 and was awarded a university medal. He is currently a research fellow in the Electromagnetics and Imaging research group of UQ. He has contributed 17 research papers to international journals and conferences and was awarded the prize for Best Student Paper at Digital Image Computing: Techniques and Applications (DICTA) 2003. His research interests include image segmentation, stereo vision, and cardiac modeling.

Hugues Talbot received the Engineering degree from Ecole Centrale de Paris in 1989, the Diplôme d'Etudes Avancée (Masters) from the Université Paris VI in 1990 and the PhD degree in mathematical morphology from Ecole Nationale Supérieure des Mines de Paris in 2003, under the guidance of professors Linn W. Hobbs (MIT), Jean Serra, and Dominique Jeulin (ENSMP). His PhD work was a collaborative work between the Isover Saint Gobain Corporation, France; MIT, Cambridge, United States; and ENSMP, France. He has been affiliated with CSIRO, Mathematical and Information Sciences, Sydney, Australia, since 1994 and is now an assistant professor at Ecole Supérieure d'Ingénieurs en Electronique et Electrotechnique, France. He has worked on a number of applied projects in image analysis with various companies and has contributed more than 50 publications to international journals and conferences. His research interests include image segmentation, linear feature analysis, texture analysis, PDEs, and algorithms.

Appendix H

Power Watersheds

This article was published in IEEE TPAMI as Couprie et al. (2011b).

```
@article{Couprie-PWSH-PAMI-2010,
Author = {Couprie, Camille and Grady, Leo and Najman, Laurent and Talbot, Hugues},
Journal = {IEEE Transactions on Pattern Analysis and Machine Intelligence},
Number = {7},
Pages = {1384 - 1399},
Title = {Power watersheds: A Unifying Graph-Based Optimization Framework},
Volume = {33},
Year = {2011}}
```

Power Watershed: A Unifying Graph-Based Optimization Framework

Camille Couprie, *Student Member*, *IEEE*, Leo Grady, *Member*, *IEEE*, Laurent Najman, and Hugues Talbot, *Member*, *IEEE*

Abstract—In this work, we extend a common framework for graph-based image segmentation that includes the graph cuts, random walker, and shortest path optimization algorithms. Viewing an image as a weighted graph, these algorithms can be expressed by means of a common energy function with differing choices of a parameter q acting as an exponent on the differences between neighboring nodes. Introducing a new parameter p that fixes a power for the edge weights allows us to also include the optimal spanning forest algorithm for watershed in this same framework. We then propose a new family of segmentation algorithms that fixes p to produce an optimal spanning forest but varies the power q beyond the usual watershed algorithm, which we term the **power watershed**. In particular, when $q = 2$, the power watershed leads to a multilabel, scale and contrast invariant, unique global optimum obtained in practice in quasi-linear time. Placing the watershed algorithm in this energy minimization framework also opens new possibilities for using unary terms in traditional watershed segmentation and using watershed to optimize more general models of use in applications beyond image segmentation.

Index Terms—Combinatorial optimization, image segmentation, graph cuts, random walker, shortest paths, optimal spanning forests, Markov random fields.

1 Introduction

Graph-based segmentation algorithms have become quite popular and mature in recent years. The modern variations on graph-based segmentation algorithms are primarily built using a small set of core algorithms—graph cuts (GC), random walker (RW), and shortest paths (SP), which are reviewed shortly. Recently, these three algorithms were all placed into a common framework that allows them to be seen as instances of a more general seeded segmentation algorithm with different choices of a parameter q [80]. In addition to these algorithms, the ubiquitous watershed segmentation algorithm [12] shares a similar seeding interface, but only recently was a connection made between the watershed algorithm and graph cuts [28]. In this paper, we show how this connection between watershed and graph cuts can be used to further generalize the seeded segmentation framework of [80] such that watershed, graph cuts, random walker, and shortest paths may all be seen as special cases of a single general seeded segmentation algorithm. Our more general formulation has several consequences which form our contributions.

1. This more general formulation reveals a previously unknown family of segmentation algorithms which we term *power watershed*. In this paper, we give an algorithm for solving the energy minimization problem associated with the power watershed and demonstrate that this new algorithm has the speed of the standard watershed but performs almost as well as or better than all of the other algorithms on our benchmark segmentation tests.
2. Placing watershed in the same framework as graph cuts, random walker, and shortest paths allows us to easily incorporate data (unary) terms into conventional watershed segmentation.
3. By placing the watershed algorithm in the same generalized framework as graph cuts, random walker, and shortest paths, it is possible to take advantage of the vast literature on improving watershed segmentation to also improve these other segmentation approaches.
4. Defining an energy function for the watershed optimization allows us to provide an MRF interpretation for the watershed.
5. By incorporating unary terms, we can push watershed beyond image segmentation into the area of general energy minimization algorithms which could be applied to any number of applications for which graph and MRF models have become standard.

Before proceeding to the exposition of our technique, we first review the graph-based segmentation literature in more detail.

- *C. Couprie, L. Najman, and H. Talbot are with the Laboratoire d'Informatique Gaspard-Monge, Université Paris-Est, Equipe A3SI, ESIEE Paris, 2, boulevard Blaise Pascal, Cité DESCARTES BP 99, 93160 Noisy-le-Grand, France.*
  E-mail: {C.Couprie, L.Najman, H.Talbot}@esiee.fr.
- *L. Grady is with the Siemens Corporate Research, Department of Imaging and Visualization, 755 College Road East, Princeton, NJ 08540.*
  E-mail: Leo.Grady@siemens.com.

Manuscript received 21 Nov. 2009; revised 31 May 2010; accepted 21 Aug. 2010; published online 9 Nov. 2010.
Recommended for acceptance by P. Felzenszwalb.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-11-0774.
Digital Object Identifier no. 10.1109/TPAMI.2010.200.

2 A Short Review of Graph-Based Segmentation

The algorithms that are reviewed in this section view the image as a graph with each pixel corresponding to a node and edges weighted to reflect changes in image intensity, color, or other features.

2.1 Watershed

There exist many possible ways for defining a watershed [88], [67], [71], [11], [28], [29]. Intuitively, the watershed of a function (seen as a topographical surface) is composed of the locations from which a drop of water could flow toward different minima. The framework allowing the formalization and proof of this statement is the *optimal spanning forest relative to the minima* [27], [28]. For the purpose of seeded image segmentation, the gradient of the image can be considered as a relief map and, instead of minima, seeds may be placed by the user or found automatically to specify the segmentation of the image into desired regions. If the gradient is inverted, the maxima are considered instead of minima, and a thalweg is computed instead of watershed. A thalweg is the deepest continuous line along a valley. In the rest of the paper, we use by convention the term "watershed" instead of "thalweg."

A maximum spanning forest (MSF) algorithm computes trees spanning all of the nodes of the graph, each tree being connected to exactly one connected seed component and the weight of the set of trees being maximum. If the seeds correspond to the maxima, the segmentation obtained by MSF is a watershed [28]. An optimal spanning forest can be computed by Kruskal's or Prim's algorithm [52], [70] among others in quasi-linear time. In Kruskal's algorithm, the edges are sorted by decreasing edge weight and chosen in that order to be added to the forest if they do not create cycles or join trees that are connected to different maxima.

Watersheds are widely used in image segmentation because there exist numerous and efficient algorithms that are easy to implement. However, segmentation results from watershed may suffer from leaks and degeneracy of the solution on the plateaus of the weight function.

2.2 Graph Cuts

The labeling produced by the GC algorithm is determined by finding the minimum cut between the foreground and background seeds via a maximum flow computation. The original work on GC for interactive image segmentation was produced by Boykov and Jolly [17], and this work has been subsequently extended by several groups to employ different features [14] or user interfaces [72], [57]. Although GC is relatively new, the use of minimal surfaces in segmentation has been a common theme in computer vision for a long time [36], [15], [63] and other boundary-based user interfaces have been previously employed [62], [33], [22], [41]. Two concerns in the literature about the original GC algorithm are metrication error ("blockiness") and the shrinking bias. Metrication error was addressed in subsequent work on GC by including additional edges [19], by using continuous max flows [7] or total variation [85]. These methods for addressing metrication error successfully overcome the problem, but may incur greater memory and computation time costs than the application of maximum flow on a 4-connected lattice. The shrinking bias can cause overly small object segments because GC minimizes boundary length. Although some techniques have been proposed for addressing the shrinking bias [19], [7], [86], these techniques all require additional parameters for computation.

2.3 Random Walker

The RW algorithm [39] is also formulated on a weighted graph and determines labels for the unseeded nodes by assigning the pixel to the seed for which it is most likely to send a random walker. This algorithm may also be interpreted as assigning the unlabeled pixels to the seeds for which there is a minimum diffusion distance [23], as a semi-supervised transduction learning algorithm [31] or as an interactive version of normalized cuts [77], [43]. Additionally, popular image matting algorithms based on quadratic minimization with the Laplacian matrix may be interpreted as employing the same approach for grouping pixels, albeit with different strategies to determine the edge weighting function [54]. Diffusion distances avoid segmentation leaking and the shrinking bias, but the segmentation boundary may be more strongly affected by seed location than with graph cuts [80].

2.4 Shortest Paths (Geodesics)

The shortest path algorithm assigns each pixel to the foreground label if there is a shorter path from that pixel to a foreground seed than to any background seed, where paths are weighted by image content in the same manner as with the GC and RW approaches. This approach was recently popularized by Bai and Sapiro [10], but variants of this idea have appeared in other sources [30], [4], [32]. The primary advantage of this algorithm is speed and prevention of a shrinking bias. However, it exhibits stronger dependence on the seed locations than the RW approach [80], is more likely to leak through weak boundaries (since a single good path is sufficient for connectivity), and exhibits metrication artifacts on a 4-connected lattice.

All of the above models may be considered as addressing energies comprising only unary and pairwise (binary) energy terms. However, recent literature has found that the addition of energy terms defined on higher order cliques can help improve performance on a variety of tasks [49], [50]. Although we do not address higher order cliques specifically in this work, we note that all recent progress in this area has been through an equivalent construction of pairwise terms. Therefore, our results could also be useful in that context. Despite the recent popularity of energies defined on higher order cliques, pairwise terms (and watershed) are still used ubiquitously in the computer vision literature and any improvement to these models can have a broad impact.

An earlier conference version of this work appeared in [24].

3 A Unifying Energy Minimization Framework

We begin our exposition by reviewing the unity framework of [80] before showing how to further broaden this

framework to provide a general seeded segmentation scheme that includes the maximum spanning forest algorithm for watershed as a special case. Examination of the special cases of this general algorithm reveals a new class of watershed segmentation models. We prove several theoretical properties of this new class of watershed and then give an algorithm for minimizing the energy associated with this generalized watershed model.

3.1 A Review of the Existing Generalized Segmentation Framework

In this section, we review the segmentation framework introduced by Sinop and Grady in [80]. A graph consists of a pair $G = (V, E)$ with vertices $v \in V$ and edges $e \in E \subseteq V \times V$ with cardinalities $n = |V|$ and $m = |E|$. An edge, e, spanning two vertices, v_i and v_j, is denoted by e_{ij}. In image processing applications, each pixel is typically associated with a node and the nodes are connected locally via a 4 or 8-connected lattice. A weighted graph assigns a real value to each edge called a weight. In this work, the weights are assumed to be nonnegative. The weight of an edge e_{ij} is denoted by $w(e_{ij})$ or w_{ij}. We also denote w_{Fi} and w_{Bi} as the unary weights penalizing foreground and background affinity at node v_i. In the context of segmentation and clustering applications, the weights encode nodal affinity such that nodes connected by an edge with high weight are considered to be strongly connected and edges with a low weight represent nearly disconnected nodes. One common choice for generating weights from image intensities is to set

$$w_{ij} = \exp(-\beta(\nabla I)^2), \tag{1}$$

where ∇I is the normalized gradient of the image I. The gradient for a gray level image is $I_i - I_j$. Details on the parameters used are given in the experimental section. We use w to denote the vector of $\mathbb{R}^m$ that contains the weights w_{ij} of every edge e_{ij} in G.

The generalized energy proposed in [80] is given by

$$\min_x \sum_{e_{ij} \in E} (w_{ij}|x_i - x_j|)^q + \sum_{v_i \in V} (w_i|x_i - y_i|)^q, \tag{2}$$

where y represents a measured configuration and x represents the target configuration. In this equation, w_{ij} can be interpreted as a weight on the gradient of the target configuration such that the first term penalizes any unwanted high-frequency content in x and essentially forces x to vary smoothly within an object while allowing large changes across the object boundaries. The second term enforces fidelity of x to a specified configuration y, w_i being weights enforcing that fidelity.

For an image segmentation in two classes, given foreground F and background B seeds, (2) may be included in the following algorithm:

$$\begin{aligned} \text{Step 1} : x = \arg\min_x & \sum_{e_{ij} \in E} (w_{ij}|x_i - x_j|)^q \\ & + \sum_{v_i} (w_{Fi}|x_i|)^q + \sum_{v_i} (w_{Bi}|x_i - 1|)^q, \\ \text{s.t. } & x(F) = 1, \;\; x(B) = 0, \\ \text{Step 2} : s_i = 1 \text{ if } x_i \geq \frac{1}{2}, & 0 \text{ if } x_i < \frac{1}{2}. \end{aligned} \tag{3}$$

In other words, we are looking for an optimum $x^\star$ of (3) that may be interpreted as a probability for a given pixel to belong to either the foreground or the background, the final decision (hard segmentation) s giving the segmentation being taken by a threshold.

It was shown in [80] that graph cuts gives a solution to this model when $q = 1$, random walker gives the solution to this model when $q = 2$, and shortest paths (geodesics) gives a solution to this model as $q \to \infty$. The case of this model with a fractional q was optimized in [79] via reweighted least squares and it was shown that intermediate values of q allowed for an algorithm which "interpolated" between the graph cuts, random walker, or shortest paths algorithms.

In related work, Strang showed in [81] that minimization of the ℓ_p norm of the gradients of a potential field with boundary conditions (in continuous space with real-valued potentials) also leads to (continuous) max flow (for an ℓ_1 norm of the gradients), the Dirichlet problem (for an ℓ_2 norm), and shortest paths (for an ℓ_∞ norm). Therefore, the framework of [80], which we now extend, may be seen as presenting similar ideas defined on an arbitrary graph, using the bridge between continuous PDEs and graph theory provided by discrete calculus [42].

3.2 Broadening the Framework to Watershed

We now broaden the segmentation algorithm in (3) to include watershed simply by separating the exponent on the weights and the variables. Specifically, we introduce parameter p to define a new segmentation model as

$$\begin{aligned} \text{Step 1} : x = \arg\min_x & \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q \\ & + \sum_{v_i} w_{Fi}^p |x_i|^q + \sum_{v_i} w_{Bi}^p |x_i - 1|^q, \\ \text{s.t. } & x(F) = 1, \;\; x(B) = 0, \\ \text{Step 2} : s_i = 1 \text{ if } x_i \geq \frac{1}{2}, & 0 \text{ if } x_i < \frac{1}{2}. \end{aligned} \tag{4}$$

As before, the final segmentation s is being chosen via a threshold.

We observe that (4) can be formulated in a general manner by rewriting it as the minimization of a general energy function $E_{p,q}(x)$ by introducing auxiliary nodes (see [44] for more details):

$$\min_x \lambda \sum_{e_{ij} \in E} \underbrace{w_{ij}^p |x_i - x_j|^q}_{\text{smoothness terms}} + \sum_{v_i \in V} \underbrace{w_i^p |x_i - y_i|^q}_{\text{data fidelity terms}}. \tag{5}$$

For example, the unary term $w_{B_i}^p |x_i - 1|^q$ can also be rewritten as $w_i^p |x_i - y_i|^q$, where y_i is an auxiliary node and the signal at this auxiliary node is fixed at $y_i = 1$.

As with (3), when p is a small finite value, then the various values of q may be interpreted, respectively, as the

TABLE 1
Relationship of Algorithms to Parameter Choices in (4)

q \ p	0	finite	∞
1	Collapse to seeds	Graph cuts	Power watershed $q=1$
2	ℓ_2 norm Voronoi	Random walker	Power watershed $q=2$
∞	ℓ_1 norm Voronoi	ℓ_1 norm Voronoi	Shortest Path Forest

Our generalized scheme for image segmentation includes several popular segmentation algorithms as special cases of the parameters p and q. The power watershed is previously unknown in the literature, but may be optimized efficiently with a maximum spanning forest calculation.

graph cuts ($q=1$) and random walker ($q=2$) algorithms. When q and p converge toward infinity with the same speed, then a solution to (4) can be computed by the shortest path (geodesics) algorithm. Those three algorithms form the underpinning for many of the advanced image segmentation methods in the literature.

It was shown in [2], [3] that when $q=1$ (graph cuts) and $p \to \infty$, then the solution of (4) is given by a maximum spanning forest algorithm. Said differently, as the power of the weights increases to infinity, then the graph cuts algorithm produces a segmentation corresponding to a segmentation by maximum spanning forest. Interpreted from the standpoint of the Gaussian weighting function in (1), it is clear that we may associate $\beta = p$ to understand that the watershed equivalence comes from operating the weighting function in a particular parameter range. An important insight from this connection is that *above some value of β, we can replace the expensive max-flow computation with an efficient maximum spanning forest computation*. By raising $p \to \infty$ and varying the power q, we obtain a previously unexplored family of segmentation models which we refer to as **power watershed**. An important advantage of power watershed with varying q is that the main computational burden of these algorithms depends on an MSF computation, which is extremely efficient [21]. In the next sections, we explore two cases that are, to the best of our knowledge, unexplored. First, we show that case p finite, $q \to \infty$ corresponds to a Voronoi diagram computation from the seeds. Second, we prove that when q is finite, as $p \to \infty$, there exists a value of p after which any of the algorithms (regardless of q) may be computed via an MSF. We then give an algorithm to minimize (4) for any value of q when $p \to \infty$. Table 1 gives a reference for the different algorithms generated by various value of p and q.

3.3 The Case p Finite, $q \to \infty$: Voronoi Diagram

Intuitively, we see that when the power over the neighboring differences tends toward infinity, the weights become negligible so that the problem obtained from (4) is a Voronoi diagram of the seeds.

A proof showing that solving the minimization problem (4) when $p = q$ and $q \to \infty$ can be achieved by shortest path computations is given in [80]. Here, we use the same idea to prove that the problem (4) in the case p finite, $q \to \infty$, is equivalent to a Voronoi diagram problem.

As $\sqrt[q]{\cdot}$ is monotonic, minimizing $E_{p,q}$ is equivalent to minimizing $\sqrt[q]{E_{p,q}}$.

First, we may factorize the objective function of our problem (4):

$$\sqrt[q]{\sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q} = \sqrt[q]{\sum_{e_{ij}\in E} \left(w_{ij}^{\frac{p}{q}} |x_i - x_j|\right)^q}. \quad (6)$$

Taking the limit $\lim_{q\to\infty} \sqrt[q]{\sum_i X_i^q}$ of a q-norm yields the maximum norm $\max_i X_i$.

Therefore, our objective function may be written as

$$\lim_{q\to\infty} \sqrt[q]{\sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q} = \lim_{q\to\infty} \max_{e_{ij}\in E} w_{ij}^{\frac{p}{q}} |x_i - x_j|. \quad (7)$$

The minimization problem can be written as

$$\begin{aligned} &\min_x \max_{e_{ij}\in E} \lim_{q\to\infty} w_{ij}^{\frac{p}{q}} |x_i - x_j|, \\ &\text{s.t. } x(F) = 1, \;\; x(B) = 0. \end{aligned} \quad (8)$$

When $q \to \infty$ and p is finite, $\frac{p}{q}$ converges toward 0, so $w_{ij}^{\frac{p}{q}}$ converges toward 1 for every edge of E. Also, for the case p finite, $q \to \infty$, can be brought back to the case $p = 0$, $q \to \infty$, whose solution is a Voronoi diagram with an ℓ_1 norm (due to the assumed 4-connectivity of the lattice).

3.4 The Case q Finite, $p \to \infty$ Leading to Watershed

We now generalize the link between GC and MSF established by Allène et al. [2], [3] by proving that GC, RW, and generally all cuts resulting out of the minimization of $E_{p,q}$ converge to MSF cuts as p tends toward infinity under the condition that all the maxima of the weight function are seeded.

The following properties are presented in the special case of segmentation into two classes, given two sets of labeled nodes F and B. However, the following results generalize easily to multilabel segmentation:

Definition 1 (q-cut). *In a graph G, let F and B be two disjoint nonempty sets of nodes, p and q two real positive values, and s the segmentation result defined in (4). The set of edges e_{ij} such that $s_i \neq s_j$ is a q-cut.*

Let Y be a subgraph of G. We say that Y is an *extension* of $F \cup B$ if each connected component of Y contains exactly one vertex of $F \cup B$ and each vertex of $F \cup B$ is contained in a connected component of Y. Consequently, it is possible to define a *label* l on each vertex of Y, 0 to the vertices connected to a vertex of B, and 1 to the vertices connected to a vertex of F.

Examples of extensions appear in Fig. 1, where F and B are displayed in (a), and two possible extensions in bold in (b) and (c), with their corresponding labels.

Let $\mathcal{F}$ be a subgraph of G. We say that $\mathcal{F}$ is a *spanning forest* (*relative to $F \cup B$*) if:

1. $\mathcal{F}$ is an extension of $F \cup B$,
2. $\mathcal{F}$ contains no cycles, and
3. $V(\mathcal{F}) = V$ ($\mathcal{F}$ is spanning all vertices of G).

The *weight* $w_{\mathcal{F}}$ of a forest $\mathcal{F}$ for w is the sum of the weight of all edges belonging to $\mathcal{F}$: $w_{\mathcal{F}} = \sum_{e_{ij}\in\mathcal{F}} w_{ij}$.

Definition 2 (MSF, MSF cut). *We say that a spanning forest $\mathcal{F}$ is an* MSF for w *if the weight of $\mathcal{F}$ is maximum*, i.e., *greater or equal to the weight of any other spanning forest.*

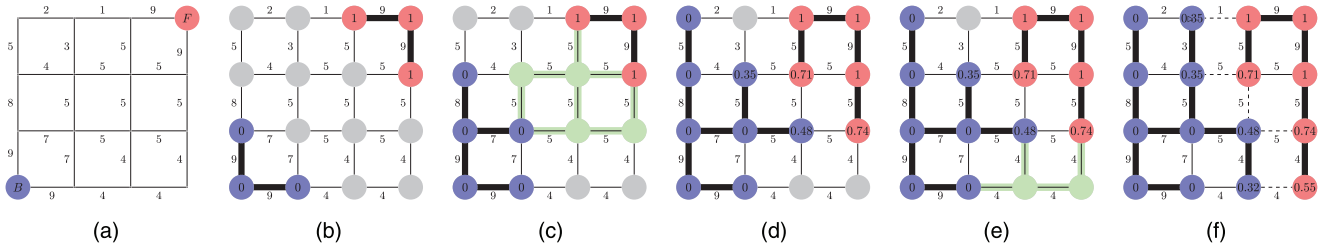


Fig. 1. Illustration of different steps in the proof of Theorem 1 for $q = 2$. The values on the nodes correspond to x, their color to s. The bold edges represent edges belonging to an MSF. (a) A weighted graph with two seeds, all maxima of the weight function are seeded, (b) first step, the edges of maximum weight are added to the forest, (c) after several steps, the next largest edge set belongs to a plateau connected to two labeled trees, (d) minimize (4) on the subset (considering the merged nodes as a unique node) with $q = 2$ (i.e., solution of the combinatorial Dirichlet problem), (e) another plateau connected to three labeled vertices is encountered, and (f) final solutions x and s obtained after a few more steps. The q-cut, which is also an MSF cut, is represented in dashed lines.

Let $\mathcal{F}$ be an MSF for w, and l its associated label. An MSF cut for w *is the set of edges e_{ij} such that $l_i \neq l_j$.*

We call a subgraph M a *maximum* of w if M is connected, all the edges of M have the same weight w_M, and the weight of any edge adjacent to M is strictly lower than w_M.

Finally, a *plateau* is a subgraph of G consisting of a maximal set of nodes connected with edges having the same weight.

Those definitions are compatible with the watershed cut framework of [28]. We may now introduce a general link between the MSF segmentation result and the solution of the optimization of (4) when the power of the weights converges toward infinity.

Theorem 1. *Let M be the subgraph of G composed of the union of all maxima of the weight function w. If every connected component of M contains at least a vertex of $B \cup F$ and $q \geq 1$, then any q-cut when $p \to \infty$ is an MSF cut for w.*

Proof. The proof is based on the construction of a set of edges that belong to the q-cut when $p \to \infty$. During the construction, we consider the edges of E in decreasing order, following Kruskal's algorithm for maximum spanning forest construction. At the end of the construction, the q-cut obtained is an MSF cut for w. The successive steps of the proof are illustrated in an example in Fig. 1.

At each step, we consider the set $E_{\max}$ of edges of maximum weight $w_{\max}$. We normalize all the weights by dividing them by $w_{\max}$, to obtain all the weights between 0 and 1 with the normalized weight of $E_{\max}$ equal to 1. The energy to minimize is also

$$\sum_{e_{ij} \in E} \left(\frac{w_{ij}}{w_{\max}} \right)^p |x_i - x_j|^q, \quad \text{s.t.} \begin{cases} x(F) = 1, \\ x(B) = 0. \end{cases} \tag{9}$$

As all maxima of the weight function contain seeds, each connected component of $E_{\max}$ has at least one labeled vertex. For every connected component $C_{\max}$ of $E_{\max}$, two cases are possible:

If $C_{\max}$ contains no vertices of different labels, the edges of weight $w_{\max}$ cannot be a part of the minimum q-cut energy when p tends toward infinity because all the other normalized weights converge toward 0 and so does any finite sum of these weights. Choosing $x_i = x_j$ for all edges $e_{ij} \in C_{\max}$ is the only possibility to eliminate the terms of maximum weight of (9). The edges of $C_{\max}$ are not included in the q-cut, and also do not belong to the MSF cut as they have to be merged to labeled nodes to form an MSF (e.g., Fig. 1b).

If $C_{\max}$ contains vertices of different labels, any labeling can be done on the plateau, because adding edges of $C_{\max}$ to the q-cut or not will always give an MSF cut on the plateau (e.g., Figs. 1c and 1d).

Repeating the steps recursively until all of the vertices are labeled, we find that in building a q-cut, we are also building an MSF cut for w in exactly the same manner as with Kruskal's algorithm. □

In Theorem 1, the condition for seeds to be the maxima of the weight function is necessary as shown in Fig. 2.

We can note that if the weights are all different, the MSF cut is unique and Theorem 1 is also true without the condition for seeds to be the maxima of the weight function.

The next property states that when the power on the neighboring node differences is strictly greater than one, the minimization of $E_{p,q}$ admits a unique solution.

Property 1. *If q is a real number such that $1 < q < \infty$, then the solution x to problem (4) is unique.*

Proof. Let A be the incidence matrix of the graph G and x a vector of $\mathbb{R}^n_+$. We note by $|\cdot|$ the elementwise absolute value operator. The function $g : x \to Ax$ is convex. The function $h : x \to |x|^q$ is convex and nondecreasing. The function $f : x \to w^T x$ is also convex and nondecreasing. Note that $E_{p,q}(x)$ can be written in the following way:

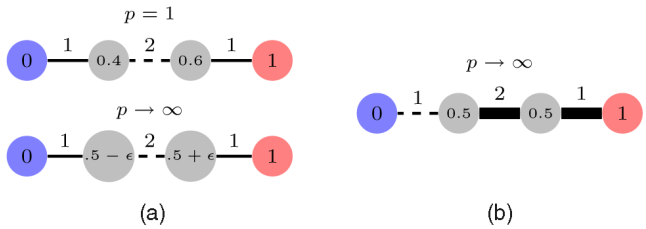


Fig. 2. Let $x = \arg\min E_{pq}$. When the maxima of the weight function are not seeded, the threshold of $\lim_{p\to\infty} x$ can be different from the limit $\lim_{p\to\infty}$ of the threshold of x. (a) Labeling $x = \arg\min E_{pq}$ and corresponding q-cut ($q = 2$) when the weights are at the power $p = 1$ and below for an arbitrary big value of p. The q-cut in dashed line remains in the center. (b) Labeling $x = \arg\min \lim_{p\to\infty} E_{pq}$ and cut (dashed) corresponding to the threshold of x. In this example, the q-cut is not an MSF cut, justifying the condition in Theorem 1. Note that the threshold of $\lim_{p\to\infty} x$ is an MSF cut, as stated later in Property 2. (a) q-cuts for different p. (b) Cut on $\lim_{p\to\infty} x$.

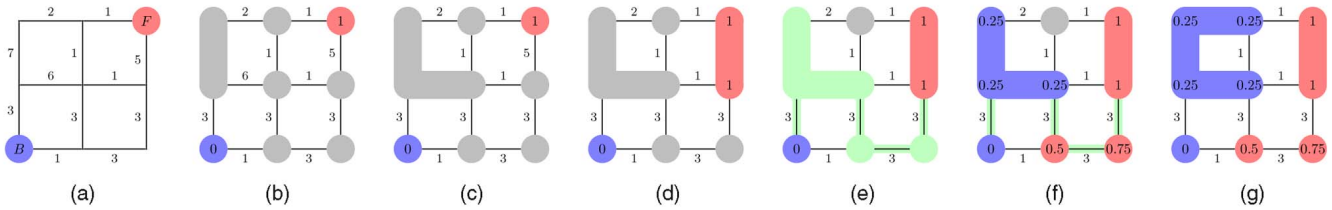


Fig. 3. Example of the behavior of the power watershed algorithm for $q = 2$ with the formation of a plateau that was not present in the original graph. (a) Initialization: A weighted graph with two seeds. (b), (c), (d) First steps: The nodes of edges of maximum weight are merged. (e) The next largest edge set belongs to a plateau connected to two different labels. (f) Minimize (4) on the subset with $q = 2$ (i.e., utilize the random walker algorithm on the plateau). (g) Final segmentation obtained after one more step.

$$E_{p,q}(x) = f \circ h \circ g(x) = w^{pT}|Ax|^q. \tag{10}$$

As h is a nondecreasing convex function and g is convex, $h \circ g$ is convex. As f is a nondecreasing convex function, $E_{p,q}$ is convex.

If $1 < q < \infty$, the function $h \circ g$ is strictly convex, so $E_{p,q}$ is a strictly convex function, and hence the minimization of $E_{p,q}$ subject to the boundary constraints is achieved by a unique x. □

Before introducing in Section 4 an algorithm to compute the solution x to the optimization of $E_{p,q}$ when $p \to \infty$, we present an interpretation of the minimization of our general energy as a maximum a posteriori approximation.

3.5 Interpretation as a Markov Random Field

An optimum $x^\star$ of (5) may be interpreted as a probability for each pixel to belong to the object (as opposed to background). More rigorously, as shown in [78], this segmentation model can be viewed as an estimation of a continuous-valued MRF. By linking the watershed algorithm to this framework, it becomes possible to view the watershed algorithm as the MAP estimation of an MRF. However, note that this analysis allows us to interpret the watershed as the MAP estimate for a particular MRF, which is in contrast to previous efforts to link a probabilistic framework with the watershed (such as [6], which uses random placements of seeds to define the most probable locations of the watershed lines).

In this section, we follow the development of [78], with modifications to incorporate the power watershed.

In the interpretation as an MRF, we define the binary segmentation label s_i for node v_i as a Bernoulli random variable (i.e., $s_i = 1$ if v_i is foreground and $s_i = 0$ if v_i is background), in which the variable x_i denotes the success probability for the distribution of s_i, i.e., $p(s_i = 1|x_i)$. In this case, the success probability may be written as

$$p(s_i = 1|x_i) = \max\{\min\{x_i, 1\}, 0\} = \begin{cases} 1, \text{ if } x_i > 1, \\ x_i, \text{ if } 0 \le x_i \le 1, \\ 0, \text{ if } x_i < 0. \end{cases} \tag{11}$$

However, the generalized mean value theorem in [78] guarantees that the optimal solution $x^\star$ to (5), assuming that the auxiliary nodes have values between 0 and 1, takes its values between 0 and 1 when the weights are all positive-valued. Consequently, in our context, we may simply set $p(s_i = 1|x_i) = x_i$ without concern that x_i will be outside the interval $[0, 1]$.

Our goal is now to infer the hidden variables x_i from the image content I. The hidden variables may be estimated in a Bayesian framework by considering the posterior model

$$p(x, s|I) \propto p(x)p(s|x)p(I|s) = p(x) \prod_{v_i \in V} p(s_i|x_i) \prod_{v_i \in V} p(I_i|s_i), \tag{12}$$

in which $p(x)$ models how the parameters of the Bernoulli variables vary spatially. The spatial smoothness prior is parameterized by

$$p(x) \propto \exp\left(-\lambda \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q\right), \tag{13}$$

where $\lambda > 0$ and the weights are strictly positive.

We can estimate the marginalized MAP criterion to obtain the optimum x^* by setting

$$x^* = \arg\max_x p(x)p(I|x) = \arg\max p(x) \sum_s p(I|s)p(s|x). \tag{14}$$

Unfortunately, $\sum_s p(I|s)p(s|x)$ is not straightforward to estimate. Therefore, we assume that we can parameterize $p(I_i|x_i)$ as

$$p(I|x) \propto \exp\left(-\sum_{v_i \in V} w_{i0}^p |x_i - 0|^q - \sum_{v_i \in V} w_{i1}^p |x_i - 1|^q\right), \tag{15}$$

where $w_{i0} \ge 0$ and $w_{i1} \ge 0$, and these terms act to bias the parameters x_i toward 0 and 1. Similarly, these terms can be used to encode the user interaction (seeding) by setting a foreground seed v_i to have weights $(w_{i0}, w_{i1}) = (0, \infty)$ and a background seed to have weights $(w_{i0}, w_{i1}) = (\infty, 0)$. With this parameterization, then the MAP estimate described in (14) is equal to our energy minimization problem from (5).

While the use of binary variables (the s variable in our formulation) is more common in recent work which applies MRFs to image segmentation, our focus on estimating a real-valued parameter or variable is far from unique in the computer vision literature. For example, in Gaussian MRFs, the variables each have a Gaussian distribution and the goal is often to estimate the (real-valued) parameters of these variables (i.e., mean and/or variance). These kinds of MRFs have been applied in image segmentation and other pattern recognition applications [48], [5], [83]. Beyond Gaussian MRFs, anisotropic diffusion has been interpreted as a continuous-valued MRF [61], [51] and MRFs requiring

continuous-valued estimations have appeared in both early work on computer vision [55], [37], [35], [16] and also recently [56], [74], [75].

We now introduce an algorithm to optimize $E_{p,q}$ when $p \to \infty$, and show that the threshold s of that solution produces an MSF cut.

4 Algorithm for Optimizing the Case q Finite, $p \to \infty$

The algorithm proposed in this section may be seen as Kruskal's algorithm for maximum spanning tree with two main differences—a forest is computed in place of a tree, and the optimization

$$\min_x \sum_{e_{ij} \in \text{plateau}} |x_i - x_j|^q \tag{16}$$

is performed on the plateaus (the maximal set of nodes connected with edges of same weight). The power watershed algorithm is detailed in Algorithm 1, and an illustration of different steps on an example is given in Fig. 3.

Algorithm 1: power watershed algorithm, optimizing $p \to \infty, q \geq 1$

Data: A weighted graph $G(V, E)$ and a set of foreground F and background B seeds
Result: A **potential function** x and a labeling s associating a label to each vertex.
Set $x_F = 1$, $x_B = 0$ and all other x values as unknown.
Sort the edges of E by decreasing order of weight.
while *any node has an unknown potential* **do**
  Find an edge (or a plateau) E_{MAX} in E of maximal weight; denote by S the set of nodes connected by E_{MAX}.
  if *S contains any nodes with known potential* **then**
    Find x_S minimizing (4) (using the input value of q) on the subset S with the weights in E_{MAX} set to $w_{ij} = 1$, all other weights set to $w_{ij} = 0$ and the known values of x within S fixed to their known values. Consider all x_S values produced by this operation as known.
  else
    Merge all of the nodes in S into a single node, such that when the value of x for this merged node becomes known, all merged nodes are assigned the same value of x and considered known.
Set $s_i = 1$ if $x_i \geq \frac{1}{2}$ and $s_i = 0$ otherwise.

In Algorithm 1, the **merge** operation of a set of nodes S consists of removing the nodes in S from the graph and replacing these nodes with a single node such that any edge spanning a node in S to nodes in $\overline{S}$ now connects the merged node to the same nodes in $\overline{S}$. Additionally, in the above algorithm, the unary terms in (4) are treated as binary terms connected to phantom seeds v_F and v_B, i.e.,

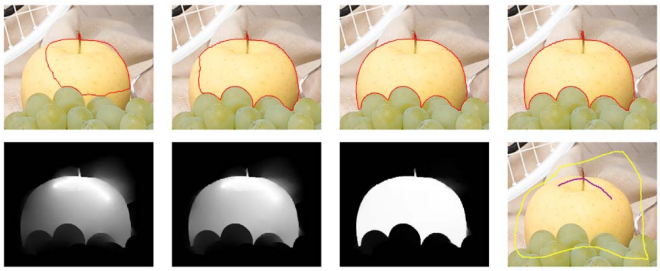


Fig. 4. Illustration of progressive convergence of the random walker result to the power watershed result as $p \to \infty$, using $q = 2$. Top row: Segmentation results obtained by random walker with weights at the power $p = 1$, $p = 8$, $p = 25$ and, finally, by the power watershed algorithm. Bottom row: Corresponding potentials for $p = 1$, $p = 8$, $p = 25$ and the input seeds.

$$\begin{aligned} &\sum_{v_i} w_{Fi}^p |x_i - 0|^q + \sum_{v_i} w_{Bi}^p |x_i - 1|^q \\ &\quad = \sum_{v_i} w_{Fi}^p |x_i - x_B|^q + \sum_{v_i} w_{Bi}^p |x_i - x_F|^q. \end{aligned} \tag{17}$$

We prove in the next section that the labeling x obtained by Algorithm 1 optimizes (4).

An illustration for this section is given in Fig. 4. The segmentation was performed with progressively larger values of p, keeping $q = 2$ and shows that the segmentation result converges to the result given by the above algorithm for the power watershed with $q = 2$. The value $q = 2$ was employed for this example since it is known that $q = 2$ forces a unique minimum to (4) regardless of the value of p.

An implementation of Algorithm 1 when $q = 2$ can be downloaded from sourceforge [1].

4.1 Justification of the Power Watershed Algorithm

We now prove that the algorithm we propose optimizes the energy presented in our framework when $q > 1$ and $p \to \infty$.

Let us define the labeling x^* as the solution $x^* = \arg\min_x E_{p,q}(x)$ defined in (4) subject to the boundary constraints. We note the labeling obtained by Algorithm 1 by $\bar{x}$.

The two following theorems, i.e., Theorems 2 and 3, state that the energy of the solution computed by the power watershed algorithm converges to the energy which minimizes $E_{p,q}$ when $p \to \infty$.

Theorem 2. *Let p, q be real positive numbers. Let w_M be the maximum weight of the graph G. For every $\delta > 0$, there exists a real k such that if $p > k$, then*

$$0 \leq \frac{E_{p,q}(\bar{x})}{{w_M}^p} - \frac{E_{p,q}(x^*)}{{w_M}^p} \leq \delta. \tag{18}$$

The proof of this theorem is given in the Appendix.

Theorem 3. *If $q > 1$, the potential x^* obtained by minimizing the energy of (4) subject to the boundary constraints converges toward the potential $\bar{x}$ obtained by Algorithm 1 as $p \to \infty$.*

Proof. We prove that by optimizing (4), we are performing the same steps as Algorithm 1. As in Theorem 1, at each step we consider a set of connected edges of maximum weight $E_{\max}$ of E, and we normalize all of the weights, also minimizing (9).

If $E_{\max}$ contains no vertices of different labels, then the weights $w_{\max}$ cannot be a part of the minimum energy when p tends toward infinity because all of the other normalized weights converge toward 0 and so does any finite sum of these weights. Choosing $x_i = x_j$ for every edge $e_{ij} = e_{\max} \in E_{\max}$ is the only possibility to eliminate the only term(s) of maximum weight of (9). This choice of $\bar{x}_i = \bar{x}_j$ is also performed by Algorithm 1 by the "merge" operation. From the standpoint of energy minimization, having $x_i = x_j$ in the graph G may be brought back to having one unique node instead of v_i, v_j, and e_{ij}. We can also replace v_i and v_j by a unique node.

If $E_{\max}$ contains vertices of a different label, as the weights of $E_{\max}$ are arbitrarily greater than the weights of the unprocessed edges, minimizing (9) boils down to minimizing

$$\sum_{e_{ij} \in E_{\max}} |x_i - x_j|^q, \tag{19}$$

with boundary conditions given by already labeled nodes. It is exactly what is performed by Algorithm 1 in the "If" part.

Repeating the steps recursively until all the vertices are labeled, we find that the Algorithm 1 procedure agrees with the energy minimization of (5). □

We can note that even if Algorithm 1 minimizes the energy $E_{p,q}$ in the case $p \to \infty$, several solutions $\bar{x}$ are possible when $q = 1$.

Property 2. *For any $q \geq 1$, the cut C defined by the segmentation s computed by Algorithm 1 is an MSF cut for w.*

Proof. At each step of Algorithm 1, we consider a set of connected edges of maximum weight $E_{\max}$.

If $E_{\max}$ contains no vertices of different labels, Algorithm 1 chooses $x_i = x_j$ for the edges $e_{ij} \in E_{\max}$. The edges of $E_{\max}$ are not included in C, and also do not belong to the MSF cut as they have to belong to an MSF since their weight is maximum.

If $E_{\max}$ contains vertices of different labels, any labeling can be done on the plateau because adding edges of $E_{\max}$ to the q-cut or not will always give an MSF cut on the plateau.

Repeating the steps of Algorithm 1 recursively until all of the vertices are labeled, we find that we are building an MSF cut for w. □

4.2 Using Mathematical Morphology for an Efficient Preprocessing Step

One difficulty in Algorithm 1 is dealing with the set of merged nodes. More precisely, when solving (16), we need to keep track of which nodes have merged (with some nodes merged multiple times). If we look informally at the "emergence" process underlying the algorithm, it will help us to locate those maximal merged nodes. Using topographical references, we view the weights as the surface of a terrain, with the weight of an edge corresponding to its altitude. If the surface were completely covered by water and the level of water slowly decreases, then islands (regional maxima) would appear that grow and merge. At a given level, when an island that does not contain a seeded pixel meets an island containing one, we can give a value to the (maximal) merged node. Indeed, we can see that any merged node consists of a connected component of an upper-level set of the weights. More precisely, let $\lambda \in \mathbb{R}^+$ and w be the weight function defined on E. We define

$$w[\lambda] = \{e \in E | w(e) \geq \lambda\}. \tag{20}$$

The graph induced by $w[\lambda]$ is called a *section* of w. A connected component of a section $w[\lambda]$ is called a *component of w (at level λ)*.

The components of w can be used to find merged nodes.

Property 3. *Any maximal merged node corresponds to a component of w that:*

- *does not contain any seed and*
- *is not contained in a larger unseeded component of w.*

Conversely, any component of w satisfying these two properties corresponds to a maximal merged node in Algorithm 1.

The components of w, ordered by the inclusion relation, form a tree called the *max-tree* [73] or the *component tree* [46], [47], [20]. Several efficient algorithms exist to compute the component tree, some quasi-linear [66] (based on union-find [84]) and some parallelized [89], [58]. From Property 3, it is easy to see how to use this tree in Algorithm 1. Note that such a tree, which keeps track of all components, can be used when one wants to improve a given segmentation result by adding extra seeds.

Another tool from mathematical morphology has been used as a preprocessing step for watershed segmentation with markers. It is called *geodesic reconstruction from the markers* [59], [12], and is given as a function w_R such that, for every edge e, we set $w_R(e)$ to be equal to the level δ of the highest component of w containing e and at least one seed node. Note that any component of w_R contains at least one seed.

Property 4. *Any maximal merged node corresponds to a connected set of edges e_{ij} that belong to a plateau of w_R and that satisfy $w_{ij} > w_R(e_{ij})$. The converse is also true.*

Property 4 also suggests that geodesic reconstruction can be used as preprocessing in Algorithm 1. Note that there exist some very efficient and easy to implement algorithms to compute a geodesic reconstruction [87], [68], [38]. Both the component tree and the geodesic reconstruction have the same theoretical complexity, so either approach could be used profitably to reduce the bookkeeping necessary to keep track of merged nodes.

Property 4 also suggests links between our framework and the classical watershed-based segmentation framework [12], [59], [60]. The framework of watershed cuts [28], [29] allows us to make a precise statement about this connection. The cut provided by a maximum spanning forest with one different seed for every maxima is called a *watershed cut*. Since geodesic reconstruction removes all maxima, which are not connected to a seed, then we can state the following:

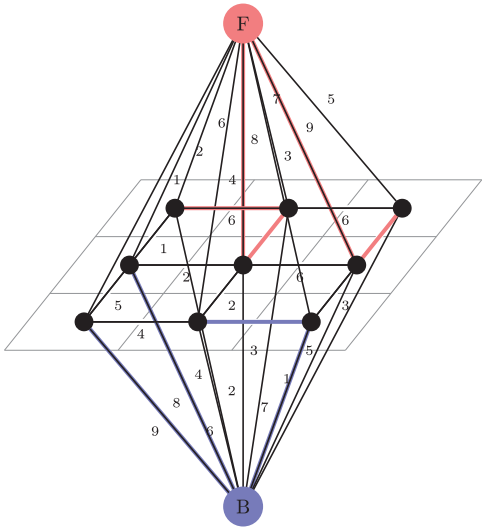


Fig. 5. Example of the unseeded segmentation of a 3×3 image computed with a maximum spanning forest (watershed).

Property 5. *Any q-cut is a watershed cut of the reconstructed weights.*

This statement ties the cuts produced by our power watershed framework to the concept of watershed cuts in the morphology literature.

4.3 Uniqueness of Solution

Most of the energy minimization problems in our framework, i.e., the cases optimized by graph cuts, shortest path forests, and maximum spanning forest algorithms (and watershed in general [12], [26], [13], [71], [34]) have the problem that the optimum solution may not be unique, for example, on plateaus. That implies that the result of each one of these algorithms depends on the implementation.

To remove such dependency, two approaches have been proposed:

- A classical approach is to compute a geodesic distance on the plateau [71] and to use that distance as a way to distinguish between points of the plateau. Generally, the cut is located on the "middle of the plateau," but other locations are possible according to the application [26], [67].
- Another proposal is the tie-zone watershed [9]; it takes into account all the possible solutions derived from a shortest-path-based watershed to generate a unique solution: When the multiple solutions disagree with each other on the segmentation result of a region (i.e., the label to be assigned), the region is included in the tie-zone and a specific tie value is assigned to each node, corresponding to the probability of assigning a label to the node according to the number of all possible assignments. A major drawback of that tie-zone approach is that nodes with equal probability of belonging to different label classes can appear.

In contrast to that approach, the power watershed computes a probability map (consisting of x in (4)) by minimizing a global energy function and, whenever q is finite and $q > 1$, the solution is unique.

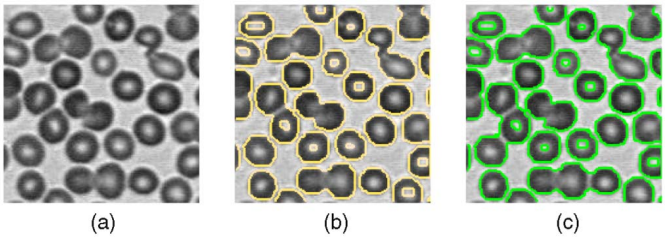


Fig. 6. Unseeded segmentation using unary terms. (a) Original image of blood cells. (b) Graph cuts. (c) MSF (watershed).

5 Results

5.1 Generality of the Framework

5.1.1 Adding Unary Terms

We now present an application of the framework to unseeded segmentation. Unary terms were first employed with graph cuts in [44]. Since this initial work, many other applications have used graph cuts with unary terms. Gathering watershed and graph cuts into the same framework allows us to employ unary terms for watershed computation.

The unary terms in (4) are treated as binary terms connected to phantom seeds v_F and v_B as in (17).

For the example of image segmentation with two labels, the weights w_{Bi} between v_B and v_i can be fixed to the absolute difference of the pixel v_i intensity with the mean of the gray scales plus the variance, and w_{Fi} to the absolute difference of the pixel v_i intensity with the mean of the gray scales minus the variance. An example of such a weighted graph is given in Fig. 5. With this construction, we can apply any of the algorithms in our framework to the resulting graph. An example of the result is shown in Fig. 6 for the purpose of segmenting blood cells. Note that those examples show how to add two phantom seeds, but this idea is extendable to more than two labels, as explained in Section 4.1.2. To the best of our knowledge, this is the first time that the watershed algorithm has been used as an unseeded segmentation method (i.e., without markers or seeds).

5.1.2 Multilabel Segmentation

Minimizing exactly the energy $E_{1,1}$ is possible by using the graph cuts algorithm in the case of two labels, but is NP-hard if constraints impose more than two different labels. However, the other algorithms presented in our framework can perform seeded segmentation with as many labels as desired efficiently.

We detail the method of multilabel segmentation in the case of the power watershed algorithm. Let N represent the number of different labels $l = 1, 2 \ldots, N$. Instead of computing an x solution of the Foreground/Background as is done for the two-labels segmentation, N solutions x^l have to be computed. In order to perform N-labels segmentation, we may define seeds at a node i by setting $x_i^l = 1$ for a given label l and $x^{\bar{l}} = 0$ for any label other than l.

The segmentation result is obtained by affecting each node v_i to the label where x_i^l is maximum:

$$s_i = \arg\max_l x_i^l. \tag{21}$$

An example of the result is shown in Fig. 7.

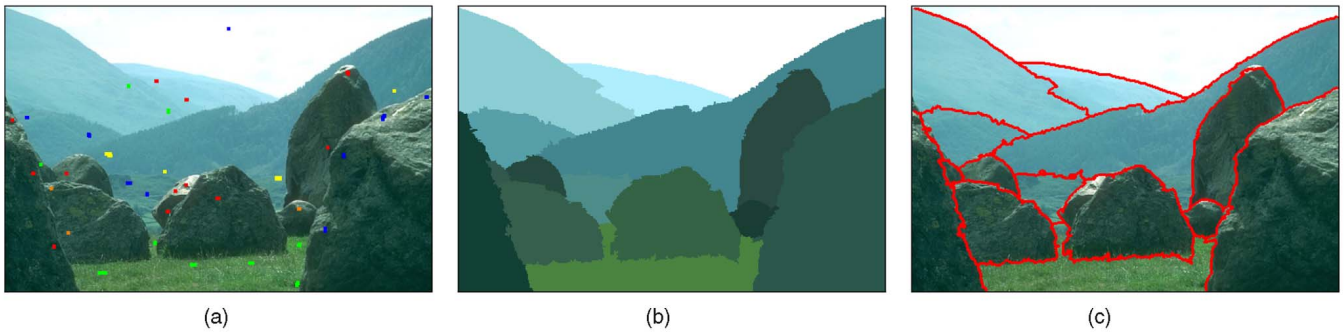


Fig. 7. Example segmentations with more than two labels. (a) Seeds. (b), (c) Power watershed result ($q = 2$).

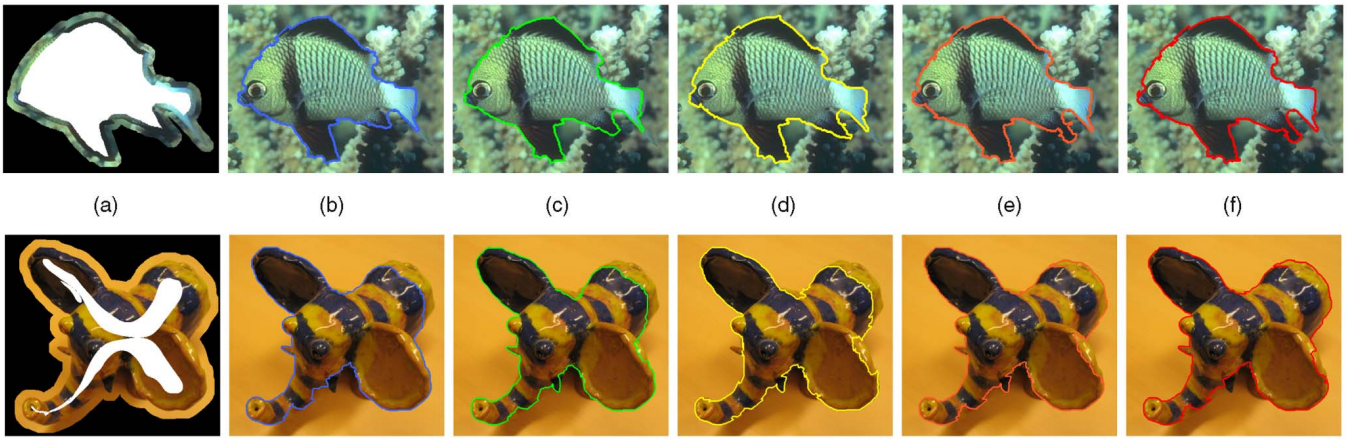


Fig. 8. Example segmentations using the provided (top images) and skeletonized (bottom images) set of seeds on the Grabcut database images: (a) Seeds, (b) graph cuts, (c) random walker, (d) shortest path, (e) maximum spanning forest (standard watershed), and (f) power watershed ($q = 2$).

5.2 Seeded Segmentation

We now demonstrate the performance of power watershed with respect to the other seeded image segmentation algorithms. In the introduction, we discussed how many of the leading graph-based segmentation algorithms (e.g., Grabcut, lazy snapping, and closed-form matting) have graph cuts, random walker, and shortest paths or watershed as an underlying component. Consequently, we will not compare the Power Watershed to any of the complete segmentation systems listed above, but rather against the comparable (component) algorithms of graph cuts, random walker, shortest paths, and watershed. Additionally, to simplify the comparison, we will not employ unary terms in our segmentations.

TABLE 2
Mean Errors on the GrabCut Database
Using Symmetically Eroded Seeds

	BE	RI	GCE	VoI	**Average rank**
Shortest paths	2.82	0.972	0.0233	0.204	**1**
Random walker	2.96	0.971	0.0234	0.204	**2.25**
MSF (Prim)	2.89	0.971	0.0244	0.209	**2.5**
Power wshed ($q = 2$)	2.87	0.971	0.0245	0.210	**3.25**
Graph cuts	3.12	0.970	0.0249	0.212	**5**

The weight parameter β was set to 600 for Graph cuts, 700 for random walker, and 900 for shortest paths in order to maximize the performances of each algorithm.

5.2.1 Quantitative Assessment

Our experiments consist of testing five algorithms embodying different combinations of p and q, consisting of GC, RW, and SP, watersheds/MSF, and power watershed using the power $q = 2$. As before, we chose to employ the power watershed algorithm with $q = 2$ due to the uniqueness of the solution to (4) for this setting.

We used the Microsoft "Grabcut" database available online [72], which is composed of 50 images provided with seeds. However, the seeds provided by the Grabcut database are generally equidistant from the ground truth boundary. To remove any bias from this seed placement on our comparative results, we produced an additional set of seeds by significantly eroding the original foreground seeds. The weights are set for all algorithms according to (1) with the value of β hand-optimized to provide the best results independently for each algorithm. As only the order

TABLE 3
Mean Errors on the GrabCut Database
Using Asymmetically Eroded Seeds

	BE	RI	GCE	VoI	**Average rank**
Graph cuts	4.70	0.953	0.0380	0.284	**1**
Power wshed ($q = 2$)	4.93	0.951	0.0407	0.297	**2.5**
Random walker	5.12	0.950	0.0398	0.294	**2.75**
MSF (Prim)	5.11	0.950	0.0408	0.298	**3.5**
Shortest paths	5.33	0.947	0.0426	0.308	**5**

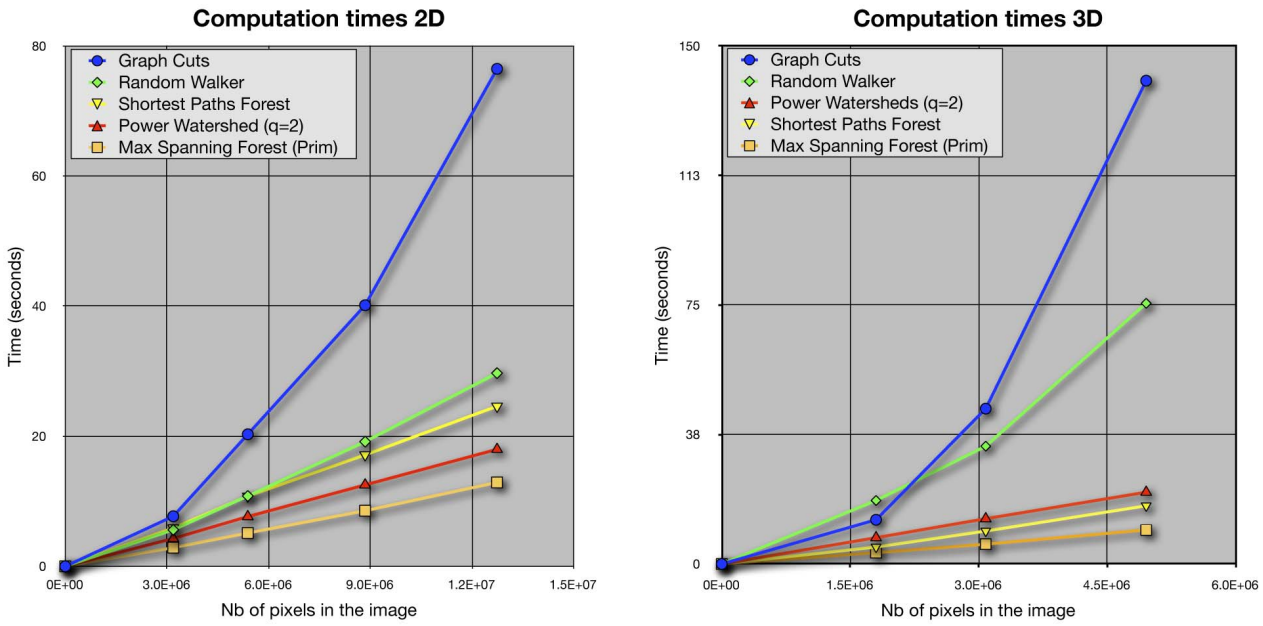


Fig. 9. Computation time for 2D and 3D seeded image segmentation. For each dimension, the times were generated by segmenting the same image scaled down.

of the weights is taken into account in the MSF and power watershed algorithms, those two algorithms are independent of β. We used the color gradient given by

$$\sqrt{\max((R_i - R_j)^2, (G_i - G_j)^2, (B_i - B_j)^2)}$$

for a color image of red, green, and blue components, R, G, B. The normalization is achieved by dividing the gradient by the maximum value of the gradient over every edge in the graph G. Example seeds and segmentations for the five algorithms with the first seeding strategy are shown at the top of Fig. 8a and with the second seeding strategy at the bottom of Fig. 8a.

Tables 2 and 3 display the performance results for these algorithms. We quantify the error in the results using four different standard segmentation measures used in [90], namely Boundary Error (BE), Rand Index (RI), Global Consistency Error (GCE), and Variation of Information (VoI). Good segmentation results are associated with low BE, high RI, low GCE, and low VoI.

When segmenting with the first seeding strategy (the seeds contained in the Grabcut database), the shortest path

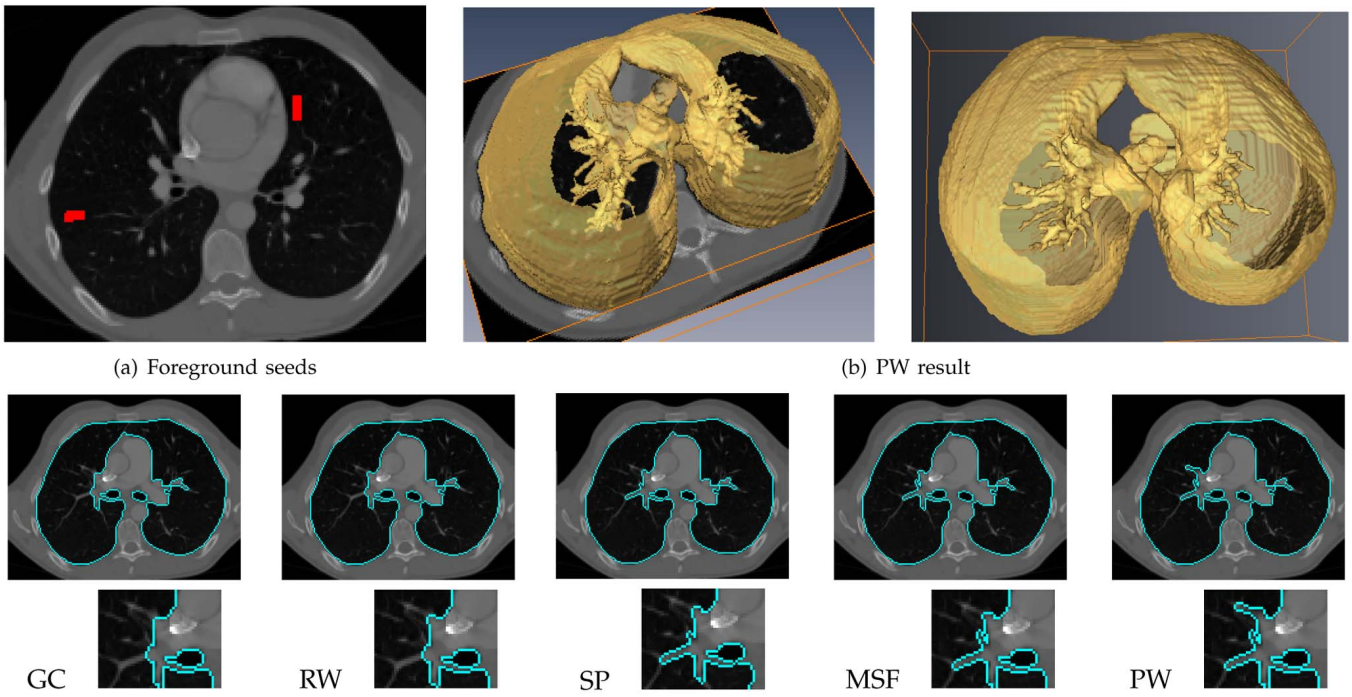


Fig. 10. Example of 3D image segmentation. The foreground seed used for this image is a small rectangle in one slice of each lung and the background seed is the frame of the image.

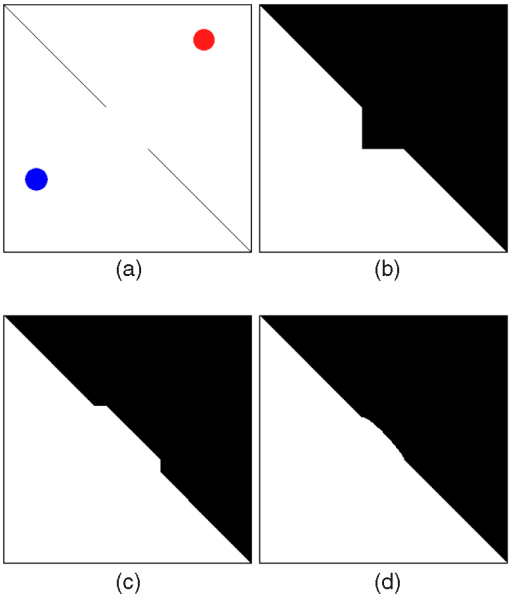


Fig. 11. (a) Image with foreground (red) and background (blue) seeds, (b) a segmentation obtained with graph cuts, (c) segmentation obtained with Prim's algorithm for maximum spanning forest, and with a shortest path algorithm, (d) segmentation obtained with random walker, as well as power watershed with $q = 2$.

algorithm is the best performer, because this algorithm does well when the seeds are placed roughly equidistant from the desired boundary [80] as they are with the first set of seeds.

The experiment on the second set of seeds shows that shortest paths are not robust to the seeds' number and centering. In fact, with this set of seeds, shortest paths is the worst performer. Graph cuts performs the best under this second seeding strategy but was the worst performer on the first one. Power watershed is in second position under the second seeding strategy, showing a good robustness to both seed quantity and location. It is interesting to note that with the first set of seeds, power watershed and maximum spanning forest results are quite similar, but with the asymetrically eroded seeds, the power watershed results outperform the standard maximum spanning forest (watershed) results. The second set of seeds contained many areas where several contours could possibly be found, given the seeds. The merging operation of the power watershed gathers undetermined areas and, in performing the random walker in these ambiguous regions, often generates a better labeling than the arbitrary labeling produced by the Prim's or Kruskal algorithms when computing the maximum spanning forest (watershed).

5.2.2 Computation Time

Computation times for segmenting 2D and 3D images using the algorithms of the framework are shown in Fig. 9. For all MSF algorithms, including the power watershed algorithm, only the order of the weights is taken into consideration for the segmentation. Also, there is no parameter choice to make for β and no exponential to take in the weight function, so it is possible to use a linear sort of the weights.

The worst-case complexity of the power watershed algorithm (obtained if all of the edges weights are equal) is given by the cost of optimizing (4) for the given q. In the best-case scenario (all weights have unique values), the power watershed algorithm has the same asymptotic complexity as the algorithm used for MSF computation, that is to say, quasi-linear. In practical applications where the plateaus have size less than some fixed value K, then the complexity of the power watershed algorithm matches the quasi-linear complexity of the standard watershed algorithm. In our experiments in Section 5 with practical image segmentation tasks, the dependence of the computation time on the image size of the power watersheds is very similar to the dependence in standard watersheds. For generating the computation time for the graph cuts algorithm, we used the software provided at http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html and described in [18]. Our implementation of the shortest path is performed with a Fibonacci heap using double precision weights. For the implementation of Prim's algorithm, weights with integer precision were used and red and black tree as a sorting data structure. Finally, the random walker algorithm was implemented following the multigrid method described in [40] for 2D image segmentation, and by a conjugate gradient descent method for 3D image segmentation. An example of 3D segmentation of a CT image of lungs is shown in Fig. 10.

Fig. 12. Example comparison of Graph cuts and Power watershed faced to a weak foreground seeds quantity. (a) Seeds (the foreground seed is in red, indicated by an arrow), (b) graph cuts segmentation result, (c) power watershed ($q = 2$) result.

5.2.3 Qualitative Assessment

Unlike most watershed algorithms, the power watershed algorithm (with $q = 2$) has the property of providing a unique segmentation. Fig. 11 shows the behavior of the algorithm of our framework in presence of a plateau. Additionally, the power watershed (with $q = 2$) is not subject to the same shrinking bias exhibited by graph cuts segmentation. Fig. 12 compares the results of graph cuts and the power watershed on an example in which the shrinking bias could substantially affect the result.

The power watershed is an MSF, and therefore it inherits the standard properties of MSF, among others, contrast invariance and scale invariance [3]. The contrast invariance property means that if a strictly monotonic transformation is applied to the weights of the graph, then the algorithm produces exactly the same result. This property is due to the fact that only the order or the weights is used to build a maximum spanning forest. The scale invariance property means that if we extend the image or graph in a way that does not change the relative ordering of weights, for example, by linear interpolation, the result is invariant.

We summarize the performance of the algorithms of the framework:

- GC is a good fit for 2D image segmentation into two labels when the seeds are far away from the boundary (asymmetric seeding), but is too slow to be used for 3D segmentation.
- SPF (geodesics) may be used if the object to segment is well-centered around foreground and background seeds.
- The RW is efficient and performs well for both seeding strategies (equidistant seeds and strongly asymmetric seeds).
- Maximum Spanning Forest (watershed) algorithms provide better segmentations than SPF when seeds are not centered, and their fast computation time makes the algorithm suitable for 3D segmentation.
- The power watershed algorithm when $q = 2$ has the additional property of a well-defined behavior in the presence of plateaus improving also the quality of the segmentation compared to standard MSF. As an MSF, it is still sensitive to leaking, but less so than traditional algorithms due to the random walk behavior. The computational speed of the power watershed is faster than all of the algorithms except the pure MSF.

6 Conclusion

In this paper, we clarified, simplified, and extended the recent work connecting graph cuts and watershed [2], [3]. Extending the framework of [80], we have proposed a general framework encompassing graph cuts, random walker, shortest path segmentation, and watersheds. This connection allowed us to define a new family of optimal spanning forests for watershed segmentation algorithms using different exponents, which we termed the "power watershed." We produced an algorithm for computing the power watershed and our experiments showed that the power watershed with $q = 2$ retains the speed of the MSF algorithm while producing improved segmentations. In addition to providing a new image segmentation algorithm, this work also showed how unary terms could be employed with a standard watershed algorithm to improve segmentation performance.

Viewed as energy minimization algorithms, graph cuts, random walker, and shortest paths have found many different applications in the computer vision field that go beyond image segmentation, such as stereo correspondence, optical flow, and image restoration (e.g., [82], [76], [53]). By placing the optimal spanning forest algorithm for watersheds in the same energy minimization framework as these other algorithms, watershed algorithms may find new uses and applications within the computer vision field beyond its traditional domain of image segmentation. Due to the relative speed of the optimal spanning forest algorithms, we believe that it may be an attractive alternative to current systems in these other applications of energy minimization.

Future work will develop along several directions. One direction is the further improvement of image segmentation algorithms using power watersheds as a component to larger systems in a similar manner as graph cuts, random walker, and shortest paths have been used. Additionally, we hope to use the common framework for these algorithms to leverage existing ideas from the watershed literature into these other algorithms. In particular, hierarchical schemes [68], [64], [65], [8], [45] look like an interesting topic that can take advantage of the power watershed uniqueness. A second direction for future work will be to characterize the limits of the watershed algorithm as an energy minimization procedure [25]. Ultimately, we hope to employ power watersheds as a fast, effective alternative to the energy minimization algorithms that currently pervade the wide variety of applications in computer vision.

Appendix

Proof of Theorem 2.

$$\begin{aligned}\frac{E_{p,q}(x)}{{w_M}^p} &= \sum_{e_M} |x_i - x_j|^q + \sum_{e_{ij} \neq e_M} \left(\frac{w_{ij}}{w_M}\right)^p |x_i - x_j|^q.\\ \frac{E_{p,q}(\bar{x})}{{w_M}^p} - \frac{E_{p,q}(x^*)}{{w_M}^p} &= \sum_{e_M} |\bar{x}_i - \bar{x}_j|^q - \sum_{e_M} |x_i^* - x_j^*|^q \\ &+ \sum_{e_{ij} \neq e_M} \left(\frac{w_{ij}}{w_M}\right)^p |\bar{x}_i - \bar{x}_j|^q - \sum_{e_{ij} \neq e_M} \left(\frac{w_{ij}}{w_M}\right)^p |x_i^* - x_j^*|^q.\end{aligned} \tag{22}$$

The first part of (22) is bounded by 0, i.e.,

$$\sum_{e_M} |\bar{x}_i - \bar{x}_j|^q - \sum_{e_M} |x_i^* - x_j^*|^q \leq 0, \tag{23}$$

because the energy obtained with $\bar{x}$ cannot be greater than the one obtained by the optimal solution x^*. More precisely, if there are no plateaus with different labels, the x_i^*, x_j^* computed on the edges e_M with Algorithm 1 are equal, leading to a sum equal to 0. Else (if there are plateaus with different labels), $\sum_{e_M} |x_i - x_j|^q$ subject to the boundary constraints is minimized on the plateaus, so the solution is optimal.

The last part of (22) is also negative, i.e.,

$$-\sum_{e_{ij} \neq e_M} \left(\frac{w_{ij}}{w_M}\right)^p |x_i^* - x_j^*|^q \leq 0. \tag{24}$$

It only remains to bound the middle part of (22),

$$\sum_{e_{ij} \neq e_M} \left(\frac{w_{ij}}{w_M}\right)^p |\bar{x}_i - \bar{x}_j|^q \leq \sum_{e_{ij} \neq e_M} \left(\frac{w_{ij}}{w_M}\right)^p \leq M_2 \left(\frac{w_{M_2}}{w_M}\right)^p, \tag{25}$$

with M_2, the number of edges of weight inferior to w_M, and w_{M_2}, the second maximum weight.

Thus, we have

$$\frac{E_{p,q}(\bar{x})}{{w_M}^p} - \frac{E_{p,q}(x^*)}{{w_M}^p} \leq M_2 \left(\frac{w_{M_2}}{w_M}\right)^p, \tag{26}$$

$$p \geq k = \frac{\log \frac{\delta}{M_2}}{\log \frac{w_{M_2}}{w_M}}. \tag{27}$$

□

ACKNOWLEDGMENTS

The authors would like to thank Jean Cousty for the numerous interesting and fruitful discussions.

REFERENCES

[1] http://sourceforge.net/projects/powerwatershed/, 2010.
[2] C. Allène, J.-Y. Audibert, M. Couprie, J. Cousty, and R. Keriven, "Some Links between Min Cuts, Optimal Spanning Forests and Watersheds," *Proc. Seventh Int'l Symp. Math. Morphology,* vol. 2, pp. 253-264, 2007.
[3] C. Allène, J.-Y. Audibert, M. Couprie, and R. Keriven, "Some Links between Extremum Spanning Forests, Watersheds and Min-Cuts," *Image and Vision Computing,* 2009.
[4] C.V. Alvino, G.B. Unal, G. Slabaugh, B. Peny, and T. Fang, "Efficient Segmentation Based on Eikonal and Diffusion Equations," *Int'l J. Computer Math.,* vol. 84, no. 9, pp. 1309-1324, 2007.
[5] A. Anandkumar, L. Tong, and A. Swami, "Detection of Gauss-Markov Random Field on Nearest-Neighbor Graph," *Proc. Int'l Conf. Acoustics, Speech and Signal Processing,* vol. 3, pp. 829-832, 2007.
[6] J. Angulo and D. Jeulin, "Stochastic Watershed Segmentation," *Proc. Eighth Int'l Symp. Math. Morphology,* pp. 265-276, 2007.
[7] B. Appleton and H. Talbot, "Globally Optimal Surfaces by Continuous Maximal Flows," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 1, pp. 106-118, Jan. 2006.
[8] P.A. Arbeláez and L.D. Cohen, "A Metric Approach to Vector-Valued Image Segmentation," *Int'l J. Computer Vision,* vol. 69, no. 1, pp. 119-126, 2006.
[9] R. Audigier and R. Lotufo, "Uniquely-Determined Thinning of the Tie-Zone Watershed Based on Label Frequency," *J. Math. Imaging Vision,* vol. 27, no. 2, pp. 157-173, 2007.
[10] X. Bai and G. Sapiro, "A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1-8, 2007.
[11] G. Bertrand, "On Topological Watersheds," *J. Math. Imaging and Vision,* vol. 22, nos. 2/3, pp. 217-230, 2005.
[12] S. Beucher and F. Meyer, "The Morphological Approach to Segmentation: The Watershed Transformation," *Math. Morphology in Image Processing,* E.R. Dougherty, ed., pp. 433-481, CRC, 1993.
[13] A. Bieniek and A. Moga, "An Efficient Watershed Algorithm Based on Connected Components," *Pattern Recognition,* vol. 33, no. 6, pp. 907-916, 2000.
[14] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive Image Segmentation Using an Adaptive GMMRF Model," *Proc. European Conf. Computer Vision,* pp. 428-441, 2004.
[15] A. Blake and A. Zisserman, *Visual Reconstruction.* MIT Press, 1987.
[16] C. Bouman and K. Sauer, "A Generalized Gaussian Image Model for Edge-Preserving MAP Estimation," *IEEE Trans. Image Processing,* vol. 2, no. 3, pp. 296-310, July 1993.
[17] Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 105-112, 2001.
[18] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1124-1137, Sept. 2001.
[19] Y. Boykov and V. Kolmogorov, "Computing Geodesics and Minimal Surfaces via Graph Cuts," *Proc. IEEE Int'l Conf. Computer Vision,* vol. 1, pp. 26-33, 2003.
[20] E.J. Breen and R. Jones, "Attribute Openings, Thinnings, and Granulometries," *Computer Vision and Image Understanding,* vol. 64, no. 3, pp. 377-389, 1996.
[21] B. Chazelle, "A Minimum Spanning Tree Algorithm with Inverse-Ackermann Type Complexity," *J. ACM,* vol. 47, no. 6, pp. 1028-1047, 2000.
[22] L.D. Cohen and R. Kimmel, "Global Minimum for Active Contour Models: A Minimal Path Approach," *Int'l J. Computer Vision,* vol. 24, no. 1, pp. 57-78, 1997.
[23] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, and S.W. Zucker, "Geometric Diffusions as a Tool for Harmonic Analysis and Structure Definition of Data: Diffusion Maps," *Proc. Nat'l Academy of Sciences USA,* vol. 102, no. 21, pp. 7426-7431, 2005.
[24] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power Watersheds: A New Image Segmentation Framework Extending Graph Cuts, Random Walker and Optimal Spanning Forest," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 731-738, Sept. 2009.
[25] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Anisotropic Diffusion Using Power Watersheds," *Proc. Int'l Conf. Image Processing,* pp. 4153-4156, 2010.
[26] M. Couprie, L. Najman, and G. Bertrand, "Quasi-Linear Algorithms for the Topological Watershed," *J. Math. Imaging Vision,* vol. 22, nos. 2/3, pp. 231-249, 2005.
[27] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed Cuts," *Proc. Seventh Int'l Symp. Math. Morphology,* vol. 1, pp. 301-312, 2007.
[28] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 8, pp. 1362-1374, Aug. 2009.
[29] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed Cuts: Thinnings, Shortest-Path Forests and Topological Watersheds," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 5, pp. 925-939, May 2010.
[30] A. Criminisi, T. Sharp, and A. Blake, "GeoS: Geodesic Image Segmentation," *Proc. European Conf. Computer Vision,* pp. 99-112, 2008.
[31] O. Duchenne, J. Audibert, R. Keriven, J. Ponce, and F. Ségonne, "Segmentation by Transduction," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[32] A.X. Falcão, R.A. Lotufo, and G. Araujo, "The Image Foresting Transformation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 1, pp. 19-29, Jan. 2004.
[33] A.X. Falcão, J.K. Udupa, S. Samarasekera, S. Sharma, B.H. Elliot, and R. de A. Lotufo, "User-Steered Image Segmentation Paradigms: Live Wire and Live Lane," *Graphical Models and Image Processing,* vol. 60, no. 4, pp. 233-260, 1998.
[34] A.X. Falcao, J. Stolfi, and R. de A. Lotufo, "The Image Foresting Transform: Theory, Algorithms, and Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 1, pp. 19-29, Jan. 2004.
[35] D. Geman and G. Reynolds, "Constrained Restoration and the Discovery of Discontinuities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 3, pp. 367-383, Mar. 1992.
[36] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 6, pp. 721-741, Nov. 1984.
[37] S. Geman and D. McClure, "Statistical Methods for Tomographic Image Reconstruction," *Proc. 46th Session Int'l Statistical Inst. Bull.,* vol. 52, pp. 4-21, Sept. 1987.
[38] T. Géraud, H. Talbot, and M. Van Droogenbroeck, "Algorithms for Mathematical Morphology," *Math. Morphology: From Theory to Applications,* L. Najman and H. Talbot, eds., pp. 345-373, Wiley-ISTE, 2010.
[39] L. Grady, "Random Walks for Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 11, pp. 1768-1783, Nov. 2006.
[40] L. Grady, "A Lattice-Preserving Multigrid Method for Solving the Inhomogeneous Poisson Equations Used in Image Analysis," *Proc. European Conf. Computer Vision,* D. Forsyth, P. Torr, and A. Zisserman, eds., pp. 252-264, 2008.
[41] L. Grady, "Minimal Surfaces Extend Shortest Path Segmentation Methods to 3D," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 2, pp. 321-334, Feb. 2010.
[42] L. Grady and J.R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science.* Springer, 2010.
[43] L. Grady and A.K. Sinop, "Fast Approximate Random Walker Segmentation Using Eigenvector Precomputation," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[44] D.M. Greig, B.T. Porteous, and A.H. Seheult, "Exact Maximum A Posteriori Estimation for Binary Images," *J. Royal Statistical Soc.,* vol. 51, no. 2, pp. 271-279, 1989.
[45] L. Guigues, J.P. Cocquerez, and H.L. Men, "Scale-Sets Image Analysis," *Int'l J. Computer Vision,* vol. 68, no. 3, pp. 289-317, 2006.
[46] P. Guillataud, "Contribution l'Analyse Dendroniques des Images," PhD thesis, Univ. de Bordeaux I, 1992.
[47] P. Hanusse and P. Guillataud, "Sémantique des Images par Analyse Dendronique," *Proc. 8ème Reconnaissance des Formes et Intelligence Artificielle,* vol. 2, pp. 577-588, 1992.

[48] J.P. Kaufhold, "Energy Formulations of Medical Image Segmentations," PhD thesis, Boston Univ., 2000.
[49] P. Kohli, M.P. Kumar, and P. Torr, "P3 & Beyond: Solving Energies with Higher Order Cliques," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2007.
[50] P. Kohli, L. Ladicky, and P. Torr, "Robust Higher Order Potentials for Enforcing Label Consistency," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[51] K. Krajsek and H. Scharr, "Diffusion Filtering without Parameter Tuning: Models and Inference Tools," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.
[52] J. Kruskal, "On the Shortest Spanning Tree of a Graph and the Traveling Salesman Problem," *Proc. Am. Math. Soc.,* vol. 7, pp. 48-50, 1956.
[53] V.S. Lempitsky, S. Roth, and C. Rother, "Fusionflow: Discrete-Continuous Optimization for Optical Flow Estimation," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[54] A. Levin, D. Lischinski, and Y. Weiss, "A Closed Form Solution to Natural Image Matting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 2, pp. 228-242, Feb. 2008.
[55] E. Levitan and G. Herman, "A Maximum A Posteriori Probability Expectation Maximization Algorithm for Image Reconstruction in Emission Tomography," *IEEE Trans. Medical Imaging,* vol. 6, no. 3, pp. 185-192, Sept. 1987.
[56] Y. Li and D.P. Huttenlocher, "Learning for Optical Flow Using Stochastic Optimization," *Proc. European Conf. Computer Vision,* pp. 379-391, 2008.
[57] Y. Li, J. Sun, C. Tang, and H. Shum, "Lazy Snapping," *Proc. ACM SIGGRAPH,* pp. 303-308, 2004.
[58] P. Matas, E. Dokládalova, M. Akil, T. Grandpierre, L. Najman, M. Poupa, and V. Georgiev, "Parallel Algorithm for Concurrent Computation of Connected Component Tree," *Proc. Int'l Conf. Advanced Concepts for Intelligent Vision Systems,* pp. 230-241, Oct. 2008.
[59] F. Meyer and S. Beucher, "Morphological Segmentation," *J. Visual Comm. and Image Representation,* vol. 1, no. 1, pp. 21-46, Sept. 1990.
[60] F. Meyer and L. Najman, "Segmentation, Minimum Spanning Tree and Hierarchies," *Math. Morphology: From Theory to Applications,* L. Najman and H. Talbot, eds., chapter 9, pp. 255-287, Wiley-ISTE, 2010.
[61] H.S. Michael, M.J. Black, and H.W. Haussecker, "Image Statistics and Anisotropic Diffusion," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 840-847, 2003.
[62] E. Mortensen and W. Barrett, "Interactive Segmentation with Intelligent Scissors," *Graphical Models and Image Processing,* vol. 60, no. 5, pp. 349-384, 1998.
[63] D. Mumford and J. Shah, "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems," *Comm. Pure and Applied Math.,* vol. 42, pp. 577-685, 1989.
[64] L. Najman, "Ultrametric Watersheds," *Proc. Ninth Int'l Symp. Math. Morphology,* M. Wilkinson and J. Roerdink, eds., pp. 181-192, Aug. 2009.
[65] L. Najman, "Ultrametric Watersheds: A Bijection Theorem for Hierarchical Edge-Segmentation," CoRR abs/1002.1887, 2010.
[66] L. Najman and M. Couprie, "Building the Component Tree in Quasi-Linear Time," *IEEE Trans. Image Processing,* vol. 15, no. 11, pp. 3531-3539, Nov. 2006.
[67] L. Najman and M. Schmitt, "Watershed of a Continuous Function," *Signal Processing,* special issue on math. morphology, vol. 38, pp. 99-112, 1994.
[68] L. Najman and M. Schmitt, "Geodesic Saliency of Watershed Contours and Hierarchical Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 12, pp. 1163-1173, Dec. 1996.
[69] L. Najman and H. Talbot, *Mathematical Morphology: From Theory to Applications.* Wiley-ISTE, 2010.
[70] R. Prim, "Shortest Connection Networks and Some Generalizations," *Bell System Technology J.,* vol. 36, pp. 1389-1401, 1957.
[71] J. Roerdink and A. Meijster, "The Watershed Transform: Definitions, Algorithms, and Parallellization Strategies," *Fundamenta Informaticae,* vol. 41, pp. 187-228, 2000.
[72] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut'—Interactive Foreground Extraction Using Iterated Graph Cuts," *Proc. ACM SIGGRAPH,* pp. 309-314, 2004.
[73] P. Salembier, A. Oliveras, and L. Garrido, "Anti-Extensive Connected Operators for Image and Sequence Processing," *IEEE Trans. Image Processing,* vol. 7, no. 4, pp. 555-570, Apr. 1998.
[74] K.G.G. Samuel and M.F. Tappen, "Learning Optimized MAP Estimates in Continuously-Valued MRF Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.
[75] U. Schmidt, Q. Gao, and S. Roth, "A Generative Perspective on MRFs in Low-Level Vision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.
[76] R. Shen, I. Cheng, X. Li, and A. Basu, "Stereo Matching Using Random Walks," *Proc. Int'l Conf. Pattern Recognition,* 2008.
[77] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.
[78] D. Singaraju, L. Grady, A.K. Sinop, and R. Vidal, "Continuous Valued MRFs for Image Segmentation," *Advances in Markov Random Fields for Vision and Image Processing,* A. Blake, P. Kohli, and C. Rother, eds., MIT Press, 2010.
[79] D. Singaraju, L. Grady, and R. Vidal, "P-Brush: Continuous Valued MRFs with Normed Pairwise Distributions for Image Segmentation," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition.* June 2009.
[80] A.K. Sinop and L. Grady, "A Seeded Image Segmentation Framework Unifying Graph Cuts and Random Walker Which Yields a New Algorithm," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.
[81] G. Strang, "l^1 and l^∞ Approximation of Vector Fields in the Plane," *Proc. US-Japan Seminar Nonlinear Partial Differential Equations in Applied Science,* pp. 273-288, 1982.
[82] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 6, pp. 1068-1080, June 2008.
[83] M.F. Tappen, C. Liu, E.H. Adelson, and W.T. Freeman, "Learning Gaussian Conditional Random Fields for Low-Level Vision," *Proc. IEEE Computer Vision and Pattern Recognition,* pp. 1-8, 2007.
[84] R. Tarjan, "Efficiency of a Good but Not Linear Set Union Algorithm," *J. ACM,* vol. 22, pp. 215-225, 1975.
[85] M. Unger, T. Pock, D. Cremers, and H. Bischof, "TVSeg—Interactive Total Variation Based Image Segmentation," *Proc. British Machine Vision Conf.,* 2008.
[86] S. Vicente, V. Kolmogorov, and C. Rother, "Graph Cut Based Image Segmentation with Connectivity Priors," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[87] L. Vincent, "Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms," *IEEE Trans. Image Processing,* vol. 2, no. 2, pp. 176-201, Apr. 1993.
[88] L. Vincent and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 6, pp. 583-598, June 1991.
[89] M.H.F. Wilkinson, H. Gao, W.H. Hesselink, J.-E. Jonker, and A. Meijster, "Concurrent Computation of Attribute Filters on Shared Memory Parallel Machines," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 10, pp. 1800-1813, Oct. 2008.
[90] A. Yang, J. Wright, Y. Ma, and S. Sastry, "Unsupervised Segmentation of Natural Images via Lossy Data Compression," *Computer Vision and Image Understanding,* vol. 110, no. 2, pp. 212-225, May 2008.

Camille Couprie received the engineering degree from ESIEE Paris, majoring in computer science and graduating with highest honors, and the master's degree in imaging science from the Université Paris Est, graduating with honors in 2008. She is currently working toward the PhD degree supported by the French Direction Générale de l'Armement MRIS Program and the Centre National de la Recherche Scientifique. Her interests include image segmentation, filtering, combinatorial optimization techniques, PDEs, and mathematical morphology. Other interests include stereovision, image registration, medical imaging, and applied topology. She is a student member of the IEEE.

Leo Grady received the BSc degree in electrical engineering from the University of Vermont in 1999 and the PhD degree from the Cognitive and Neural Systems Department, Boston University in 2003. Since Autumn 2003, he has worked as a principal research scientist at Siemens Corporate Research (Princeton, New Jersey) in the Image Analytics and Informatics Department. His research interests include image segmentation, data clustering, biomedical imaging, complex networks, learning, compressive sensing and filtering using techniques from graph theory, discrete calculus, optimization, and PDEs. Recent work has been devoted to writing a book *Discrete Calculus: Applied Analysis on Graphs for Computational Science* (Springer, 2010), which focuses on how the structure of connections between data points may be used to improve the analysis and understanding of the data. He is a member of the IEEE and the IEEE Computer Society.

Laurent Najman received the habilitation diriger les recherches from the University of Marne-la-Valle, the engineering degree from the Ecole Nationale Supérieure des Mines de Paris in 1991, and the PhD degree in applied mathematics from the Université Paris-Dauphine in 1994 with the highest honor (Felicitations du Jury). He worked in the central research laboratories of Thomson-CSF for three years. After completing the engineering degree, he worked on infrared image segmentation problems using mathematical morphology. Then, he joined a start-up company named Animation Science in 1995 as director of research and development. The particle systems technology for computer graphics and scientific visualization developed by the company under his technical leadership received several awards, including the "European Information Technology Prize 1997" awarded by the European Commission (Esprit program) and by the European Council for Applied Science and Engineering, as well as the "Hottest Products of the Year 1996" awarded by *Computer Graphics World*. In 1998, he joined OCÉ Print Logic Technologies as a senior scientist. There, he worked on various image analysis problems related to scanning and printing. In 2002, he joined the Informatics Department of ESIEE, Paris, where he is currently a professor and member of the Gaspard-Monge Computer Science Research Laboratory (LIGM), Université Paris-Est. His current research interests include discrete mathematical morphology.

Hugues Talbot received the PhD degree in mathematical morphology from the Ecole Nationale Supérieure des Mines de Paris (ENSMP) with highest honors in 1993, under the guidance of professors Linn W. Hobbs (MIT), Jean Serra, and Dominique Jeulin (ENSMP). He was affiliated with CSIRO, Mathematical and Information Sciences, Sydney, Australia, between 1994 and 2004. He is currently a computer science professor at ESIEE, affiliated with the Université Paris-Est, France. He has worked on a number of applied projects in image analysis with various companies, earning the Australian Institute of Engineers Award in 2004 and the DuPont Australian and New Zealand Innovation Award in 2005 for work related to melanoma diagnosis. He has contributed more than 80 publications to international journals and conferences. His research interests include image segmentation, optimization, and algorithms with application to image analysis and computer vision. He is a member of the IEEE and the IEEE Computer Society.

LICENSE

List of Figures

1.1 A computer vision problem: outlining a kangaroo 14
1.2 Un problème d'analyse d'image 15
1.3 An image including several thin objects 15
1.4 Details of figure 1.3 : two fibres that are less visible.. 16
1.5 A thresholding of figure 1.3 16
1.6 Scale and local orientation. (a) shows some text, where thin objects are made of the strokes composing letters ; (b) shows the same text blurred by a Gaussian kernel of standard deviation $\sigma = 5$. Here thin objects are the words themselves. (c) shows again the same text, blurred with $\sigma = 10$. Thin objects have all but disappeared. 17
1.7 Discrete cylinders and their scale-dependent local orientation. Orientations near the boundary of the cylinders are normal to the surface, while the orientations near the axis of symmetry of the cylinders are oriented along the axis. 18
1.8 Image of neurites 18
1.9 Segmentation and identification of neurites in HCA-image. 19
1.10 A 3D view of an *in-vivo* moth neurite. 19
1.11 A image of the eye fundus showing the blood vessels. 20
1.12 An image of a stent guide in X-ray fluorescence imaging. 21
1.13 Classical test colour image of Barbara, with oriented textures. 21
1.14 Most modern digital cameras use a colour filter array (CFA) like the one in (a), using a single sensor but causing only one colour channel to be captured at each pixel. Interpolation of nearby pixels must be performed to recover the other two channels. However using simple interpolation schemes causes artefacts on thin objects as in (b), (c). 22
1.15 Some frames of the "table tennis" CIF test video sequence, originally meant for video coding. 23
1.16 Some frames from the Zhang Yimou feature film "House of Flying Daggers" (`http://www.imdb.com/title/tt0385004/`). Some scratches and artifacts are obvious and appear in a single frame (see (b)). Others, like on the forehead of the heroine, only appear when viewing consecutive frames. 23
1.17 Reinforcement fibres in a substrate. Slices through the material (a) and (b) and associated segmentation (c) and (d). 24
1.18 A projection view of the heart coronaries in CT injected numerical angiogram in (a). In (b), a current state-of-the-art segmentation of the main arteries. . . . 25

1.19 A so-called "corrosion cast" of the blood vessels in the brain. The level of details required to image this structure in 3D *in vivo* is beyond the capabilities of current imaging devices. . . . 26
1.20 Cracks developing in concrete through fast drying: (a) Initial image at t=0 ; (b) cracks at t=24h ; (c) 3D rendering of the crack network. Data courtesy CNRS-ICMCB. . . . 27
1.21 Seismic data rendering, courtesy Petrobras . . . 27

2.1 Eigenvalue analysis of the 3D Hessian . . . 33
2.2 An image of fingerprints and their orientation . . . 36
2.3 Gabor wavelets in the real domain . . . 38
2.4 Filter bank from Malik et al. (2001):. . . . 38
2.5 Two overlapping circles in orientation-space . . . 40
2.6 Contourlet filter bank . . . 41
2.7 Example of contourlet decomposition . . . 41
2.8 Conceptual implementation of the Ridgelet Transform . . . 42
2.9 Conceptual implementation of first-gen curvelets . . . 43
2.10 Second-generation curvelet . . . 44
2.11 Geometric flow an directional bandelets . . . 45
2.12 Principle of non-local means . . . 46
2.13 The classical "burner" images and processing . . . 48
2.14 Why the Van-Herk algorithm leads to loss of translation-invariance . . . 48
2.15 Vincent's local shortest paths . . . 49
2.16 Fibre segmentation with local shortest paths. . . . 50
2.17 An image of thin man-made vitreous fibres, including thin ones. . . . 52
2.18 Images of cauliflowers . . . 53
2.19 Line filtering sensitivity in the presence of noise . . . 54
2.20 Line filtering with rank-max opening . . . 55
2.21 On an image of fibres in a microprobe (a). Comparison between translation invariant opening (b) and rank-max opening (c). Colours indicate orientation. . 56
2.22 b_1, b_2, b_3 are successors of a and a_1, a_2, a_3 are the predecessors of b. . . . 58
2.23 A set $X \subseteq E$ (black points on the left) and its opening $\alpha_6(X)$ (black points on the right). The points on the right with a white center are discarded by the transform. . . . 58
2.24 Four different adjacencies defining overlapping $\frac{\pi}{2}$ cones. . . . 59
2.25 Line filtering with incomplete path openings . . . 60
2.26 Eye fundus blood vessel analysis. (a) reference DoG ridge detection ; (b) path closing detection; (c) path closing classification; (d) ground truth. . . . 62
2.27 Path operators on 3D fibres: (a) complete operator; (b) robust operator; (c) classification. Results from Cokelaer (2013) . . . 64
2.28 X-ray fluorescence guide-wire extraction. Vesselness, Rotated Filtered Banks and Polygonal path images, from Bismuth et al. (2012). . . . 65

3.1 Embedding and evolving a curve as a level-set. . . . 78
3.2 A graph with edge weights interpreted as capacities . . . 82
3.3 Why the Random Walker copes well with weak boundaries. . . . 84
3.4 Watershed segmentation example. . . . 85

3.5 Illustration of the Power Watershed algorithm. 90
3.6 Slices of a 3D lung segmentation. 90
3.7 3D lung segmentation. 93

4.1 A very simple maxflow on a directed graph. 101
4.2 Simple binary seeded graph and associated segmentation 106
4.3 Grey-level segmentation of an apple 107
4.4 Segmentation with unary weights 108
4.5 Binary segmentation with unary weights 108
4.6 Construction of the Ishikawa optimization graph. 112
4.7 Graph-representability of unary and binary weights 116
4.8 Notations for the α-expansion graph 117
4.9 Joint quantization and denoising 119
4.10 Graph arrangement for (4.31) 120
4.11 Convergence of the Power Watershed. (a) Initial image with seeds; (b) Result of Power Watershed;(c) Result of Random walker with $p = 2$; (d) RW with $p = 16$: (e) RW with $p = 30$. 121
4.12 Comparison of Perona-Malik(PM), and power watershed(PW) algorithms for denoising a synthetic image. (b) PM used with 80 iterations $\alpha = 0.0015$, leading to a good PSNR but with remaining isolated noisy pixels. (c) PM, best compromise found for this image to remove the isolated pixels with 50 iterations and $\alpha = 0.0005$. (d) A median filtered image as initialization and $\lambda = 0.975$ allows to obtain a better SNR while removing isolated noisy pixels. 124
4.13 (a) Three dots in a 4×5 lattice. (b) Associated lattice weighted by an Euclidean distance map from the dots, with Foreground and Background seeds. (c) First steps of the power watershed algorithm optimizing (4.31) in the case $q = 2$ and $p \to \infty$. Nodes with a maximum weight are merged. (d) A plateau of weight 2 (in green) including different seeded nodes is encountered. The Random walker algorithm is applied to label the nodes on the plateau. (e,f) New plateaus of weight 1 and 0 are encountered, the Random walker algorithm is applied, (g) Final labeling x solution of (4.31). The resulting isocontour is represented in red. 125
4.14 Grid size : $234 \times 297 \times 301$, complete and zoomed results. (a) Total variation minimization result, (b) Graph cuts result, (c) power watershed result. Isosurfaces at 0.5 have been extracted on all results, and were downsampled by 2 to render the surfaces. 126

5.1 The catenoid test problem. (a) The expected minimal surface, constructed analytically. (b) The iso-surface at value 0.5 of P at convergence. (c) A horizontal slice through P. The expected cross-section is overlayed in black. (d) A vertical slice through P. The expected cross-section is overlayed in black. 142
5.2 The incidence matrix of a small graph 146
5.3 The difference between classical max-flow on a graph with the combinatorial continuous max-flow (CCMF) on a graph is that classical max-flow uses edge-weighted capacities while CCMF uses node-weighted capacities. This difference is manifest in the different solutions obtained for both algorithms and the algorithms required to find a solution. Specifically, the solution to the CCMF problem on a lattice does not exhibit metrication bias. 147

5.4 The dual problem to CCMF is a node-weighted minimum cut in which the variable λ is a weighted indicator vector labeling boundary nodes and the variable ν is a nearly binary vector indicating the source/sink regions. As a result, the contours of ν are slightly blurry. This is due to the equilibrium effect between the two dual variables. In practice, as λ is nonzero only in presence of a contour, ν is binary almost everywhere, except on a very thin line. 148
5.5 The catenoid test problem: The source is constituted by two full circles and sink by the remaining boundary of the image. (a) Surface computed analytically, (b) isosurface of P obtained by AT-CMF, (c) isosurface of ν obtained by CCMF. The root mean square error (RSME) has been computed to evaluate the precision of the results to the surface computed analytically. The RSME for AT-CMF is 1.98 and for CCMF 0.75. The difference between those results is due to the fact that the AT-CMF algorithm enforces exactly the source and sink points, leading to discretization around the disks. In contrast, the boundary localized around the seeds of ν is smooth, composed of grey levels. Thus the resulting isosurface computed by CCMF is more precise. 149
5.6 Segmentation of an artificial image with AT-CMF (top row) and CCMF (bottom row). Top row (AT-CMF): (a) Image where the black and white discs are seeds. AT-CMF result stopped in (b) after 100 iterations, (c) 1000 iterations, (d) 10000 iterations. Bottom row (CCMF): (e) Image where the black and white discs are seeds, CCMF result ν after (f) 1 iterations, (g) ν after 15 iterations iterations and (h) threshold of the final ν. 150
5.7 Local and nonlocal denoising (Gaussian noise of variance $\sigma^2 = 20$). The nonlocal weights are computed using Bresson (2009). Regularization parameters used to obtain the best results: (e) $\lambda = 0.048$ (f) $\lambda = 0.093$. 155
5.8 Denoising and deblurring an MRI image corrupted with synthetic Gaussian 5×5 blur and Gaussian noise ($\sigma^2 = 10$). 156
5.9 Example of mesh denoising using DCTV on the spatial coordinates of the nodes. We use the Root Square Mean Error to measure an average distance to the original mesh. (c) Laplacian smoothing (d) The M+SFBF algorithm was used to optimize DCTV with $\lambda = 0.25$. The Root Mean Square Error measures a distance between the result and the original mesh. 157

List of Tables

1.1 Some characteristics of thin objects 28

2.1 Possible eigenvalue responses for various shapes 32
2.2 Some reference algorithms for the segmentation and classification of blood vessels 61

3.1 Generalized schemes for image segmentation 89
3.2 Qualitative assessment of many popular segmentation methods 94

4.1 Maxflow-mincut duality formulation 100
4.2 Table of general-purpose maxflow-mincut algorithms 104
4.3 Interpretation of (4.31) depending on the values of p and q. 120

Bibliography

Adams, R. and Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647.

Allène, C., Audibert, J.-Y., Couprie, M., Cousty, J., and Keriven, R. (2007). Some links between min-cuts, optimal spanning forests and watersheds. In *Procs. 8th International Symposium on Mathematical Morphology*, Rio de Janeiro, Brazil.

Alpert, B., Beylkin, G., Coifman, R., and Rokhlin, V. (1993). Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM Journal on Scientific Computing*, 14(1):159–184.

Altendorf, H. and Jeulin, D. (2009). 3D directional mathematical morphology for analysis of fiber orientations. *Image Analysis and Stereology*, 28(143–153).

Appleton, B. (2004). *Globally minimal contours and surfaces for image segmentation*. PhD thesis, University of Queensland. http://espace.library.uq.edu.au/eserv/UQ:9759/ba_thesis.pdf.

Appleton, B. and Sun, C. (2003). Circular shortest paths by branch and bound. *Pattern Recognition*, 36(11):2513–2520.

Appleton, B. and Talbot, H. (2003). Globally optimal surfaces by continuous maximal flows. In *Digital image computing: techniques and applications*, pages 987–996.

Appleton, B. and Talbot, H. (2005a). Efficient path openings and closings. In Ronse, C., Najman, L., and Decencière, E., editors, *Mathematical Morphology: 40 Years On*, volume 30 of *Computational Imaging and Vision*, pages 33–42, Dordrecht. Springer-Verlag.

Appleton, B. and Talbot, H. (2005b). Globally optimal geodesic active contours. *Journal of Mathematical Imaging and Vision*, (23):67–86.

Appleton, B. and Talbot, H. (2006). Globally minimal surfaces by continuous maximal flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):106–118.

Ardon, R. and Cohen, L. (2006). Fast constrained surface extraction by minimal paths. *International Journal of Computer Vision*, 69(1):127–136.

Bamberger, R. and Smith, M. (1992). A filter bank for the directional decomposition of images: Theory and design. *Signal Processing, IEEE Transactions on*, 40(4):882–893.

Bauschke, H. and Combettes, P. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. Springer.

Benmansour, F. and Cohen, L. (2009). Fast object segmentation by growing minimal paths from a single point on 2d or 3d images. *Journal of Mathematical Imaging and Vision*, 33(2):209–221.

Benmansour, F. and Cohen, L. (2011). Tubular structure segmentation based on minimal path method and anisotropic enhancement. *International Journal of Computer Vision*, 92(2):192–210.

Bernard, D. et al. (2012). Assessment of concrete cracks formations. *in preparation*.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302.

Beucher, S. (1990). *Segmentation d'images et morphologie mathématique*. PhD thesis, Ecole des Mines de Paris.

Beucher, S. and Lantuéjoul, C. (1979). Use of watersheds in contour detection. In *Int. Workshop on Image Processing*, Rennes, France. CCETT/IRISA.

Bischof, L., Buckley, M., Lagerstrom, R., Sun, C., Talbot, H., Wang, D., and Vallotton, P. (2005). Image analysis of neurite branching: High-content screening at high speed. *AMERICAN BIOTECHNOLOGY LABORATORY*, 23(10):22.

Bismuth, V., Vaillant, R., Talbot, H., and Najman, L. (2012). Curvilinear structure enhancement with the polygonal path image - application to guide-wire segmentation in X-Ray fluoroscopy. In Ayache, N., Delingette, H., Golland, P., and Mori, K., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, volume 7511 of *Lecture Notes in Computer Science*, pages 9–16, Nice, France. Springer Berlin Heidelberg.

Black, M. J., Sapiro, G., Marimont, D. H., and Heeger, D. (1998). Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432.

Bloch, I. and Maitre, H. (1994). Fuzzy mathematical morphology. *Annals of Mathematics and Artificial Intelligence*, 10(1):55–84.

Bloomberg, D. and Vincent, L. (2010). Document image applications. In Najman, L. and Talbot, H., editors, *Mathematical morphology: from theory to applications*, chapter 18, pages 423–436. ISTE-Wiley.

Bougleux, S., Elmoataz, A., and Melkemi, M. (2007). Discrete regularization on weighted graphs for image and mesh filtering. In *Proceedings of International Conference on Scale Space and Variational Methods in Computer Vision*, SSVM '07, pages 128–139, Ischia, Italy.

Boykov, Y. and Funka-Lea, G. (2006). Graph cuts and efficient nd image segmentation. *International Journal of Computer Vision*, 70(2):109–131.

Boykov, Y. and Jolly, M.-P. (2002). Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, pages 105–112, Vancouver, BC , Canada.

Boykov, Y. and Kolmogorov, V. (2003). Computing geodesics and minimal surfaces via graph cuts. In *International Conference on Computer Vision*, pages 26–33, Nice, France.

Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137.

Boykov, Y., Veksler, O., and Zabih, R. (1998). Markov random fields with efficient approximations. In *CVPR*, pages 648–655.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

Bradley, S., Hax, A., and Magnanti, T. (1977). *Applied mathematical programming*. Addison Wesley.

Bredies, K., Kunisch, K., and Pock, T. (2010). Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526.

Bresenham, J. E. (1965). Algorithm for computer control of digital plotter. *IBM System Journal*, 4:25–30.

Bresson, X. (2009). A short note for nonlocal TV minimization.

Bresson, X. and Chan, T. (2008). Non-local unsupervised variational image segmentation models. Technical Report 08–67, UCLA CAM report.

Buades, A., Coll, B., and Morel, J. (2005). A review of image denoising algorithms, with a new one. *J. Multiscale Model. Simul.*, 4(2):490–530.

Buckley, M. and Talbot, H. (2000). Flexible linear openings and closings. In Vincent, L. and Bloomberg, D., editors, *Mathematical Morphology and its application to image analysis*, pages 109–118, Palo Alto. ISMM, Kluwer.

Burt, P. and Adelson, E. (1983). The laplacian pyramid as a compact image code. *Communications, IEEE Transactions on*, 31(4):532–540.

Candes, E. (1998). *Ridgelets: theory and applications*. PhD thesis, Stanford University.

Candès, E. and Donoho, D. (1999). Ridgelets: A key to higher-dimensional intermittency? *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 357(1760):2495–2509.

Candes, E. and Donoho, D. (2003). New tight frames of curvelets and optimal representations of objects with piecewise c2 singularities. *Communications on pure and applied mathematics*, 57(2):219–266.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.

Cardoso, M., Clarkson, M., Modat, M., Talbot, H., Couprie, M., and Ourselin, S. (2011). Topologically correct cortical segmentation using khalimsky's cubic complex framework. In *SPIE Medical Imaging*, page 79620P. International Society for Optics and Photonics.

Caselles, V., Kimmel, R., and Sapiro, G. (1997). Geodesic active contours. *International Journal on Computer Vision*, 22(1):61–79.

Chambolle, A. (2004). An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1–2):89–97.

Chambolle, A. and Lions, P.-L. (1997). Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188.

Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145.

Chan, T. and Bresson, X. (2010). Continuous convex relaxation methods for image processing. In *Proceedings of ICIP 2010*. Keynote talk, http://www.icip2010.org/file/Keynote/ICIP

Chan, T. and Vese, L. (2001). Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277.

Chan, T. F., Golub, G. H., and Mulet, P. (1999). A nonlinear primal-dual method for total variation based image restoration. *SIAM J. Sci. Comput.*, 20(6):1964–1977.

Chaux, C., Jezierska, A., Pesquet, J.-C., and Talbot, H. (2010). A spatial regularization approach for vector quantization. *Journal of Mathematical Imaging and Vision*, pages 1–16. 10.1007/s10851-010-0241-3.

Chen, G. and Kégl, B. (2007). Image denoising with complex ridgelets. *Pattern Recognition*, 40(2):578–585.

Chen, J., Sato, Y., and Tamura, S. (1998). Orientation space filtering for multiple orientation line segmentation. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 311–317. IEEE.

Chen, J., Sato, Y., and Tamura, S. (2000). Orientation space filtering for multiple orientation line segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(5):417–429.

Chen, Y. and Hsu, W. (1989). An interpretive model of line continuation in human visual perception. *Pattern Recognition*, 22(5):619–639.

Chouzenoux, E., Jezierska, A., Pesquet, J.-C., and Talbot, H. (2013a). A majorize-minimize subspace approach for \ell_2-\ell_0 image regularization. *SIAM Journal on Imaging Sciences*, 6(1):563–591.

Chouzenoux, E., Pesquet, J., Talbot, H., and Jezierska, A. (2011). A memory gradient algorithm for l2 - l0 regularization with applications to image restoration. In *International Conference on Image Processing (ICIP)*, Brussels, Belgium.

Chouzenoux, E., Zolyniak, F., Gouillart, E., and Talbot, H. (2013b). A majorize-minimize memory gradient algorithm applied to x-ray tomography. In *Proceedings of ICIP*, Melbourne, Australia.

Cohen, L. and Deschamps, T. (2007). Segmentation of 3d tubular objects with adaptive front propagation and minimal tree extraction for 3d medical imaging. *Computer Methods in Biomechanics and Biomedical Engineering*, 10(4):289–305.

Cohen, L. D. and Kimmel, R. (1997). Global minimum for active contour models: A minimal path approach. *International Journal of Computer Vision*, 24(1):57–78.

Cokelaer, F. (2013). *Améliorations des ouvertures par chemins pour l'analyse d'images à N dimensions et implémentations optimisées*. PhD thesis, Université de Grenoble.

Cokelaer, F., Talbot, H., and Chanussot, J. (2012). Efficient robust d-dimensional path operators. *IEEE Journal of Selected Topics in Signal Processing*, 6(7):830 –839.

Combettes, P. and Pesquet, J. (2012). Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators. *Set-Valued and Variational Analysis*, 20(2):307–330.

Combettes, P. L. and Pesquet, J.-C. (2008). A proximal decomposition method for solving convex variational inverse problems. *Inverse problems*, 24(6).

Combettes, P. L. and Pesquet, J.-C. (2010). Proximal splitting methods in signal processing. In Bauschke, H. H., Burachik, R., Combettes, P. L., Elser, V., Luke, D. R., and Wolkowicz, H., editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer-Verlag, New York.

Condat, L. (2012). A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, pages 1–20.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. MIT Press and McGraw-Hill, second edition.

Coupé, P., Hellier, P., Prima, S., Kervrann, C., and Barillot, C. (2008a). 3d wavelet subbands mixing for image denoising. *Journal of Biomedical Imaging*, 2008:1.

Coupé, P., Yger, P., Prima, S., Hellier, P., Kervrann, C., and Barillot, C. (2008b). An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images. *Medical Imaging, IEEE Transactions on*, 27(4):425–441.

Couprie, C. (2011). *Graph based variational optimization and applications in computer vision*. PhD thesis, Université Paris-Est.

Couprie, C., Bresson, X., Najman, L., Talbot, H., and Grady, L. (2011a). Surface reconstruction using power watershed. In *International Symposium on Mathematical Morphology (ISMM)*, Lake Maggiore, Italy.

Couprie, C., Grady, L., Najman, L., and Talbot, H. (2009). Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *Proceedings of ICCV 2009*, pages 731–738, Kyoto, Japan. IEEE.

Couprie, C., Grady, L., Najman, L., and Talbot, H. (2011b). Power watersheds: A unifying graph-based optimization framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1384 – 1399.

Couprie, C., Grady, L., Talbot, H., and Najman, L. (2010). Anisotropic diffusion using power watersheds. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 4153–4156, Honk-Kong.

Couprie, C., Grady, L., Talbot, H., and Najman, L. (2011c). Combinatorial Continuous Maximum Flows. *SIAM Journal on Imaging Sciences*, 4(3):905–930.

Couprie, C., Talbot, H., Pesquet, J.-C., Najman, L., and Grady, L. (2011d). Dual constrained tv-based regularization. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 945–948. IEEE.

Cousty, J., Bertrand, G., Najman, L., and Couprie, M. (2008). Watershed cuts: minimum spanning forests and the drop of water principle. *IEEE transactions on pattern analysis and machine intelligence*, pages 1362–1374.

Cousty, J., Bertrand, G., Najman, L., and Couprie, M. (2010). Watershed cuts: thinnings, shortest-path forests and topological watersheds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):925–939.

Cserti, J. (2000). Application of the lattice Green's function for calculating the resistance of an infinite network of resistors. *American Journal of Physics*, 68:896.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2006). Image denoising with block-matching and 3-D filtering. In *Proceedings of SPIE*, volume 6064, pages 354–365.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007). Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095.

Danielsson, P.-E. and Lin, Q. (2001). Efficient detection of second-degree variations in 2D and 3D images. *Journal of Visual Communication and Image Representation*, 12:255–305.

Danielyan, A., Katkovnik, V., and Egiazarian, K. (2012). Bm3d frames and variational image deblurring. *Image Processing, IEEE Transactions on*, 21(4):1715–1728.

Daragon, X., Couprie, M., and Bertrand, G. (2002). Marching chains algorithm for Alexandroff-Khalimsky spaces. In *SPIE Vision Geometry XI*, volume 4794, pages 51–62.

Darbon, J., Cunha, A., Chan, T., Osher, S., and Jensen, G. (2008). Fast non-local filtering applied to electron cryomicrography. In *ISBI*, pages 1331–1334, Paris. IEEE.

Darbon, J. and Sigelle, M. (2006a). Image restoration with discrete constrained total variation part i: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26(3):261–276.

Darbon, J. and Sigelle, M. (2006b). Image restoration with discrete constrained total variation part ii: Levelable functions, convex priors and non-convex cases. *Journal of Mathematical Imaging and Vision*, 26(3):277–291.

De Mazancourt, T. and Gerlic, D. (1983). The inverse of a block-circulant matrix. *IEEE Trans. on Antennas and Propagation*, 31(5):808–810.

Derpanis, K. and Gryn, J. (2005). Three-dimensional nth derivative of gaussian separable steerable filters. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–553. IEEE.

Dijkstra, E. (1959). A note on two problems in connexion with graph. *Numerische Mathematik*, 1:269–271.

Dinic, E. A. (1970). An algorithm for the solution of the max-flow problem with the polynomial estimation. *Doklady Akademii Nauk SSSR (in Russian. In english: Soviet Mathematics Doklady 11)*, 194(4):1277–1280.

Do, M. and Vetterli, M. (2005). The contourlet transform: an efficient directional multiresolution image representation. *Image Processing, IEEE Transactions on*, 14(12):2091–2106.

Do, M. N. (2001). *Directional multiresolution image representations.* PhD thesis, EPFL.

Dokladal, P., Lohou, C., Perroton, L., and Bertrand, G. (1999). Liver blood vessels extraction by a 3-d topological approach. In *MICCAI*, Cambridge, Great Britain.

Donoho, D. and Duncan, M. (2000). Digital curvelet transform: strategy, implementation, and experiments. In *AeroSense 2000*, pages 12–30. International Society for Optics and Photonics.

Dougherty, E. and Lotufo, R. (2003). *Hands-on morphological image processing.* SPIE press.

Douglas, J. and Rachford, H. (1956). On the numerical solution of heat conduction problems in two or three space variables. *Trans. Amer. Math. Soc*, 82:421–439.

Doyle, P. and Snell, J. (1984). Random walks and electric networks, volume 22 of Carus Mathematical Monographs. *Mathematical Association of America, Washington, DC*, 52.

E.Bayer, B. (1976). *Color Imaging Array.* United State Patent, 3,971,065.

Edmonds, J. and Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264.

Elmoataz, A., Lezoray, O., and Bougleux, S. (2008). Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *Image Processing, IEEE Transactions on*, 17(7):1047–1060.

Fadili, M., Starck, J., et al. (2007). Curvelets and ridgelets. *Encyclopedia of Complexity and Systems Science*, 3:1718–1738.

Farey, J. (1816). On a curious property of vulgar fractions. *Philosophical Magazine Series 1*, 47(217):385–386.

Felzenszwalb, P. F. and Huttenlocher, D. R. (2004). Efficient belief propagation for early vision. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision*, pages 261–268.

Fischer, B. and Modersitzki, J. (1999). Fast inversion of matrices arising in image processing. *Numerical Algorithms*, 22(1):1–11.

Fleming, W. (1954). Functions with generalized gradients and generalized surfaces. *Anali di Matematica*, 44:93–103.

Ford, J. L. R. and Fulkerson, D. R. (1962). *Flows in Networks*. Princeton University Press, Princeton, NJ.

Ford, L. R. and Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404.

Frangi, A., Niessen, W., Koen, L., and Viergever, M. (1998). Multiscale vessel enhancement filtering. *Lecture Notes in Computer Science*, 1496:130ff.

Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. *tpami*, 13(9):891–906.

Gabay, D. (1983). Applications of the method of multipliers to variational inequalities. In Fortin, M. and Glowinsk, R., editors, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, pages 299–331. North-Holland, Amsterdam.

Gabor, D. (1965). Information theory in electron microscopy. *Laboratory Investigation*, 14(6):801–807.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *pami*, 6:721–741.

Gil, J. and Kimmel, R. (2002). Efficient dilation, erosion, opening, and closing algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(12):1606–1617.

Gil, J. and Werman, M. (1993). Computing 2-d min, median and max filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):504–507.

Gilboa, G. and Osher, S. (2007). Nonlocal linear image regularization and supervised segmentation. *SIAM J. Multiscale Model. Simul.*, 6(2):595–630.

Gilboa, G. and Osher, S. (2008). Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028.

Giusti, E. (1984). *Minimal surfaces and functions of bounded variation*, volume 80. Birkhauser.

Goldberg, A. and Tarjan, R. (1988). A new approach to the maximum-flow problem. *Journal of the ACM*, 35:921–940.

Goldenberg, R., Kimmel, R., Rivlin, E., and Rudzsky, M. (2001). Fast geodesic active contours. *IEEE Trans. On Image Processing*, 10(10):1467–1475.

González, G., Aguet, F., Fleuret, F., Unser, M., and Fua, P. (2009). Steerable features for statistical 3d dendrite detection. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pages 625–632.

Grady, L. (2005). Multilabel random walker image segmentation using prior models. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:763–770.

Grady, L. (2006a). Computing exact discrete minimal surfaces: Extending and solving the shortest path problem in 3D with application to segmentation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 69–78. IEEE.

Grady, L. (2006b). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783.

Grady, L. and Funka-Lea, G. (2004). Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, pages 230–245.

Grady, L. and Polimeni, J. (2010). *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer Publishing Company, Incorporated.

Grady, L. and Schwartz, E. (2006). Isoperimetric graph partitioning for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(3):469–475.

Graf, J. and Wörn, H. (2007). Oriented median filtering preserving significant image structures. In *9th International Workshop on Computer Science and Information Technologies (CSIT2007)*, volume 2, pages 4–8.

Gratin, C. (1989). *MICROMORPH PC - Manuel de Référence - Version II*. Ecole des Mines de Paris.

Guigues, L., Cocquerez, J., and Le Men, H. (2006). Scale-sets image analysis. *International Journal of Computer Vision*, 68(3):289–317.

Gunturk, B., Glotzbach, J., Altunbasak, Y., Schafer, R., and Mersereau, R. M. (2005). Demosaicking: Color filter array interpolation. *IEEE Signal processing magazine*, 22:44–54.

Heijmans, H. (1994). *Morphological image operators*. Advances in Electronics and Electron Physics Series. Academic Press, Boston.

Heijmans, H. (1996). Morphological filters for dummies. In Maragos, P., Schafer, R. W., and Butt, M. A., editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 127–137, Atlanta, GA. proceedings for ISMM'96, Kluwer Acad.

Heijmans, H., Buckley, M., and Talbot, H. (2004). Path-based morphological openings. In *Proceedings of IEEE ICIP 2004*, pages 3085–3088, Singapore.

Heijmans, H., Buckley, M., and Talbot, H. (2005). Path openings and closings. *Journal of Mathematical Imaging and Vision*, 22:107–119.

Hendriks, C. (2010). Constrained and dimensionality-independent path openings. *Image Processing, IEEE Transactions on*, 19(6):1587 –1595.

Henriksen, J. (2007). 3d surface tracking and approximation using gabor filters. *South Denmark University (March 28, 2007)*.

Hiremath, P., Prema, T., and Sharan, B. (2011). Performance comparison of wavelet transform and contourlet transform based methods for despeckling medical ultrasound images. *International Journal of Computer Applications*, 26(9):34–41.

Horowitz, S. and Pavlidis, T. (1974). Picture segmentation by a directed split-and-merge procedure. In *Proceedings of the Second International Joint Conference on Pattern Recognition*, volume 424, page 433.

Hu, T. C. (1969). *Integer Programming and Network Flows*. Addison-Wesley, Reading, MA.

Huang, K., Zhang, D., and Wang, K. (2009). Non-local means denoising algorithm accelerated by gpu. *Source: Proceedings of the SPIE-The International Society for Optical Engineering*, 7497:749711.

Huber, P. and Ronchetti, E. (1981). *Robust statistics*, volume 1. Wiley Online Library.

Iri, M. (1979). *Survey of Mathematical Programming*. North-Holland, Amsterdam.

Ishikawa, H. (2003). Exact optimization for Markov random fields with convex priors. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336.

Ishikawa, H. and Geiger, D. (1999). Mapping image restoration to a graph problem. In *Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, pages 189–193, Antalya, Turkey.

Jacob, M. and Unser, M. (2004). Design of steerable filters for feature detection using canny-like criteria. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1007–1019.

Jacques, L., Duval, L., Chaux, C., and Peyré, G. (2011). A panorama on multiscale geometric representations, intertwining spatial, directional and frequency selectivity. *Signal Processing*, 91(12):2699–2730.

Jaquet, C., Ando, E., Viggiani, G., and Talbot, H. (2013). Estimation of separating planes between touching 3d objects using power watershed. In *Proceedings of the 11th International Symposium on Mathematical Morphology (ISMM)*, pages 452–463, Uppsala, Sweden. Springer.

Jezierska, A. (2013). *Image Restoration in the Presence of Poisson-Gaussian Noise*. PhD thesis, Université Paris-Est.

Jezierska, A., Chaux, C., Talbot, H., and Pesquet, J.-C. (2010). image quantisation under spatial smoothness constraints. In *Proceedings of ICIP 2010*, Honk-Kong. IEEE signal processing society.

Jezierska, A., Chouzenoux, E., Pesquet, J.-C., and Talbot, H. (2012). A primal-dual proximal splitting approach for restoring data corrupted with poisson-gaussian noise. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1085 –1088.

Jezierska, A., Talbot, H., Veksler, O., and Wesierski, D. (2011). A fast solver for truncated-convex priors: quantized-convex split moves. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, Saint Petersburg, Russia.

Juan, O. and Boykov, Y. (2007). Capacity scaling for graph cuts in vision. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE.

Kakutani, S. (1945). Markov processes and the Dirichlet problem. In *Proc. Jap. Acad*, volume 21, pages 227–233.

Karasaridis, A. and Simoncelli, E. (1996). A filter design technique for steerable pyramid image transforms. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 4, pages 2387–2390. IEEE.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331.

Kervrann, C., Boulanger, J., and Coupé, P. (2007). Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. *Scale Space and Variational Methods in Computer Vision*, pages 520–532.

Khalimsky, E., Kopperman, R., and Meyer, P. (1990). Computer graphics and connected topologies on finite ordered sets. *Topology Appl*, 36(1):1–17.

Kharlamov, A. and Podlozhnyuk, V. (2007). Image denoising. *NVIDIA Corporation*.

Kingsbury, N. (2001). Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, 10(3):234–253.

Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1568–1583.

Kolmogorov, V. and Rother, C. (2007). Minimizing nonsubmodular functions with graph cuts-a review. *IEEE transactions on pattern analysis and machine intelligence*, pages 1274–1279.

Kolmogorov, V. and Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(2):147–159.

Koren, I., Laine, A., and Taylor, F. (1995). Image fusion using steerable dyadic wavelet transform. In *Image Processing, 1995. Proceedings., International Conference on*, volume 3, pages 232–235. IEEE.

Kruskal, J. J. (1956). On the shortest spanning subtree of a graph and the travelling salesman problem. *Proceedings of the AMS*, 7(1).

Kumar, M. P. and Torr, P. H. S. (2008). Improved moves for truncated convex models. In *Proceedings of Advances in Neural Information Processing Systems*.

Kurdy, M. and Jeulin, D. (1989). Directional mathematical morphology operations. In *Proceedings of the 5th European Congress for Stereology*, volume 8/2, pages 473–480, Freiburg im Breisgau, Germany. Acta Stereologica.

Le Pennec, E. and Mallat, S. (2005). Sparse geometric image representations with bandelets. *Image Processing, IEEE Transactions on*, 14(4):423–438.

Lempitsky, V., Rother, C., Roth, S., and Blake, A. (2010). Fusion moves for markov random field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1392–1405.

Lermé, N., Malgouyres, F., and Létocart, L. (2010). Reducing graphs in graph cut segmentation. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 3045–3048. IEEE.

Levinshtein, A., Stere, A., Kutulakos, K., Fleet, D., Dickinson, S., and Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2290–2297.

Levitin, E. and Polyak, B. (1966). Constrained minimization methods. *U.S.S.R. Comput. Math. Math. Phys*, 6:1–50.

Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers. ISBN 0-7923-9418-6.

Liu, Y., Wang, J., Chen, X., Guo, Y., and Peng, Q. (2008). A robust and fast non-local means algorithm for image denoising. *Journal of Computer Science and Technology*, 23(2):270–279.

Mairal, J. (2010). *Sparse coding for machine learning, image processing and computer vision*. PhD thesis, Ecole Normale Supérieure de Cachan.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2009). Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE.

Malgouyres, F. (2002a). Mathematical analysis of a model which combines total variation and wavelet for image restoration. *ue Journal of information processes*, 2(1):1–10.

Malgouyres, F. (2002b). Minimizing the total variation under a general convex constraint for image restoration. *Image Processing, IEEE Transactions on*, 11(12):1450–1456.

Malik, J., Belongie, S., Leung, T., and Shi, J. (2001). Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27.

Malladi, R., Sethian, J., and Vemuri, B. (1995). Shape modelling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175.

Mallat, S. (1999). *A wavelet tour of signal processing*. Academic press.

Mallat, S. (2009). *A wavelet tour of signal processing*. Academic press, 3rd edition.

Mallat, S. and Peyré, G. (2007). A review of bandlet methods for geometrical image representation. *Numerical Algorithms*, 44(3):205–234.

Manniesing, R., Viergever, M. A., and Niessen, W. J. (2006). Vessel enhancing diffusion: A scale space representation of vessel structures. *Medical Image Analysis*, 10(6):815–825.

Már Sigurdsson, E., Valero-Valbuena, S., Benediktsson, J. A., Chanussot, J., Talbot, H., and Stefansson, E. (2013). Automatic retinal vessel extraction based on directional mathematical morphology and fuzzy classication. *in preparation*.

Marak, L. (2012). *On continuous maximum flow image segmentation algorithm*. PhD thesis, Université Paris-Est.

Marc Lebrun (2012). An Analysis and Implementation of the BM3D Image Denoising Method. *Image Processing On Line*.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, pages 187–217.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423.

Menzies, S. W., Crotty, K. A., Ingvar, C., and McCarthy, W. H. (1996). *An Atlas of Surface Microscopy of Pigmented Skin Lesions*. McGraw-Hill, Roseville, Australia. ISBN 0 07 470206 8.

Meyer, F. (1994). Topographic distance and watershed lines. *Signal Processing*, 38(1):113–125.

Meyer, F. (2004). Levelings, image simplification filters for segmentation. *Journal of Mathematical Imaging and Vision*, 20(1):59–72.

Meyer, F. and Beucher, S. (1990). Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46.

Morard, V. (2012). *Détection de structures fines par traitement d'images et apprentissage statistique : application au contrôle non-destructif.* PhD thesis, Ecole Nationale Supérieure des Mines de Paris - Paristech.

Moreau, J. (1965). Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93(2):273–299.

Mortensen, E. and Barrett, W. (1998). Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384.

Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685.

Murota, K. (2000). Algorithms in discrete convex analysis. *IEICE TRANSACTIONS ON INFORMATION AND SYSTEMS E SERIES D*, 83(3):344–352.

Murota, K. (2003). *Discrete convex analysis*. SIAM monographs on Discrete Mathematics and Applications, Philadelphia, USA.

Najman, L. (2011). On the equivalence between hierarchical segmentations and ultrametric watersheds. *Journal of Mathematical Imaging and Vision*, 40(3):231–247.

Najman, L., Cousty, J., Couprie, M., Talbot, H., Clément-Guinaudeau, S., Goissen, T., and Garot, J. (2008). An open, clinically-validated database of 3d+t cine-mr images of the left ventricle with associated manual and automated segmentation. In *ISC/NA-MIC Workshop on Open Science at MICCAI 2007*, number 157. The Insight Journal. url=`http://hdl.handle.net/1926/550`.

Najman, L. and Schmitt, M. (2002). Geodesic saliency of watershed contours and hierarchical segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(12):1163–1173.

Najman, L. and Talbot, H., editors (2010). *Mathematical Morphology: from theory to applications*. ISTE-Wiley, London, UK. ISBN 978-1848212152.

Ngo, P., Kenmochi, Y., Passat, N., and Talbot, H. (2012). Combinatorial structure of rigid transformations in 2d digital images. *Computer Vision and Image Understanding*. In print.

Nguyen, T.-A., Dufour, A., Tankyevych, O., Nakib, A., Petit, E., Talbot, H., and Passat, N. (2013). Thin structure filtering framework with non-local means, gaussian derivatives and spatially-variant mathematical morphology. In *International Conference in Image Processing (ICIP)*, Melbourne, Australia.

Nickolls, J., Buck, I., Garland, M., and Skadron, K. (2008). Scalable parallel programming with CUDA. *Queue*, 6(2):40–53.

Niemeijer, M., Staal, J., van Ginneken, B., Loog, M., and Abramoff, M. D. (2004). Comparative study of retinal vessel segmentation methods on a new publicly available database. In *Medical Imaging 2004*, pages 648–656. International Society for Optics and Photonics.

Nieuwenhuis, C., Töppe, E., and Cremers, D. (2013). A survey and comparison of discrete and continuous multi-label optimization approaches for the potts model. *International Journal of Computer Vision*, pages 1–18.

Nikolova, M., Esedoglu, S., and Chan, T. F. (2006). Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math*, 66(5):1632–1648.

Nozawa, R. (1990). Max-flow min-cut theorem in an anisotropic network. *Osaka J. Math*, 27:805–842.

Nozawa, R. (1994). Examples of max-flow and min-cut problems with duality gaps in continuous networks. *Mathematical programming*, 63(1):213–234.

Osher, S. and Sethian, J. (1988). Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, 79(1):12–49.

Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, 11:285–296.

Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Dover. ISBN 0486402584.

Paragios, N. and Deriche, R. (2002a). Geodesic active contours and level sets for the detection and tracking of moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(3):266–280.

Paragios, N. and Deriche, R. (2002b). Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247.

Perona, P. (1995). Deformable kernels for early vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(5):488–499.

Perona, P. (1998). Orientation diffusions. *IEEE Trans. Image Processing*, 7:457–467.

Peyré, G., Bougleux, S., Cohen, L., et al. (2011). Non-local regularization of inverse problems. *Inverse Problems and Imaging*, 5(2):511–530.

Peyré, G. and Mallat, S. (2008). Orthogonal bandelet bases for geometric images approximation. *Communications on Pure and Applied Mathematics*, 61(9):1173–1212.

Pham, D., Xu, C., and Prince, J. (2000). CURRENT METHODS IN MEDICAL IMAGE SEGMENTATION1. *Biomedical Engineering*, 2(1):315.

Pock, T. and Chambolle, A. (2011). Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1762–1769. IEEE.

Pock, T., Cremers, D., Bischof, H., and Chambolle, A. (2009). An algorithm for minimizing the Mumford-Shah functional. In *12th International Conference on Computer Vision*, pages 1133–1140. IEEE.

Pock, T., Cremers, D., Bischof, H., and Chambolle, A. (2010). Global solutions of variational models with convex regularization. *SIAM Journal on Imaging Sciences*, 3(4):1122–1145.

Prim, R. (1957). Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6):1389–1401.

Ronse, C. (1986). Erosion of narrow image features by combination of local low rank and max filters. In *Proc. 2nd Int. Conf. on Image Processing and its Applications*, pages 77–81.

Ronse, C. (1988). Extraction of narrow peaks and ridges in images by combination of local low rank and max filters: implementation and applications to clinical angiography. Working Document WD47, Philips Research Laboratories, Brussels, Belgium.

Rosen, B. and Vincent, L. (1994). Morphological image processing techniques applied to detection of correlogram tracks. *U.S. Navy Journal of Underwater Acoustics*, 44(2):571–586.

Rouchdy, Y. and Cohen, L. (2008). Image segmentation by geodesic voting. application to the extraction of tree structures from confocal microscope images. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–5. IEEE.

Rouchdy, Y. and Cohen, L. (2009). The shading zone problem in geodesic voting and its solutions for the segmentation of tree structures. application to the segmentation of microglia extensions. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 66–71. IEEE.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268.

Sagiv, C., Sochen, N., and Zeevi, Y. (2006). Integrated active contours for texture segmentation. *Image Processing, IEEE Transactions on*, 15(6):1633–1646.

Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., and Kikinis, R. (1998). Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical image analysis*, 2(2):143–168.

Schlesinger, D. and Flach, B. (2006). Transforming an arbitrary minsum problem into a binary one. Technical report, Dresden University.

Sedgewick, R. (2002). *Algorithms in C*. Number 5. Addison-Wesley, third edition.

Selesnick, I., Baraniuk, R., and Kingsbury, N. (2005). The dual-tree complex wavelet transform. *Signal Processing Magazine, IEEE*, 22(6):123–151.

Serra, J. and Vincent, L. (1992). An overview of morphological filtering. *Circuits Systems Signal Process*, 11(1):47–108.

Sethian, J. (1999a). Fast marching methods. *SIAM review*, 41(2):199–235.

Sethian, J. (1999b). *Level set methods and fast marching methods*. Cambridge University Press. ISBN 0-521-64204-3.

Shabou, A., Tupin, F., and Darbon, J. (2009). A graph-cut based algorithm for approximate mrf optimization. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2413–2416. IEEE.

Shan, H., Ma, J., and Yang, H. (2009). Comparisons of wavelets, contourlets and curvelets in seismic denoising. *Journal of Applied Geophysics*, 69(2):103–115.

Shih, F. and Cheng, S. (2004). Adaptive mathematical morphology for edge linking. *Information sciences*, 167(1):9–21.

Simoncelli, E. and Farid, H. (1996). Steerable wedge filters for local orientation analysis. *Image Processing, IEEE Transactions on*, 5(9):1377–1382.

Simoncelli, E., Freeman, W., Adelson, E., and Heeger, D. (1992). Shiftable multiscale transforms. *Information Theory, IEEE Transactions on*, 38(2):587–607.

Sinha, S. N., Mordohai, P., and Pollefeys, M. (2011). Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *proceedings of the 11th IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, Rio de Janeiro.

Sinop, A. K. and Grady, L. (2007). A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE.

Skladnev, V., Gutenev, A., Menzies, S., Bischof, L., Talbot, H., Breen, E., and Buckley, M. (2004). Diagnostic feature extraction in dermatological examination. WO 02/094098 A1 and US 2004/0267102 A1.

Soille, P. (2002). On morphological operators based on rank filters. *Pattern recognition*, 35(2):527–535.

Soille, P. (2008). Constrained connectivity for hierarchical image partitioning and simplification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1132–1145.

Soille, P., Breen, E., and Jones, R. (1996). Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):562–567.

Soille, P. and Talbot, H. (1998). Image structure orientation using mathematical morphology. volume 2, pages 1467–1469, Brisbane. ICPR, IEEE Computer Society.

Soille, P. and Talbot, H. (2001). Directional morphological filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1313–1329.

Staal, J., Abràmoff, M. D., Niemeijer, M., Viergever, M. A., and van Ginneken, B. (2004). Ridge-based vessel segmentation in color images of the retina. *Medical Imaging, IEEE Transactions on*, 23(4):501–509.

Starck, J., Candès, E., and Donoho, D. (2002). The curvelet transform for image denoising. *Image Processing, IEEE Transactions on*, 11(6):670–684.

Stawiaski, J., Decencière, E., and Bidault, F. (2007). Computing approximate geodesics and minimal surfaces using watershed and graph cuts. In Banon, G. J. F., Barrera, J., Braga-Neto, U. d. M., and Hirata, N. S. T., editors, *Proceedings of the 8th International Symposium on Mathematical Morphology*, volume 1, pages 349–360. Instituto Nacional de Pesquisas Espaciais (INPE).

Strang, G. (1983). Maximal flow through a domain. *Mathematical Programming*, 26:123–143.

Sun, C., Jones, R., Talbot, H., Wu, X., Cheong, K., Beare, M., Buckley, M., and Berman, M. (2006). Measuring the distance of vegetation from powerlines using stereo vision. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):269–283.

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. (2008). A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1068–1080.

Taguchi, A. and Iri, M. (1982). Continuum approximation to dense networks and its application to the analysis of urban road networks. In Cottle, R. W. et al., editors, *Applications*, volume 20 of *Mathematical Programming Studies*, pages 178–217. Springer Berlin Heidelberg. 10.1007/BFb0121231.

Talbot, H. and Appleton, B. (2007). Efficient complete and incomplete paths openings and closings. *Image and Vision Computing*, 25(4):416–425.

Talbot, H., Lee, T., Jeulin, D., Hanton, D., and Hobbs, L. W. (2000). Image analysis of insulation mineral fibres. *Journal of Microscopy*, 200(3):251–268.

Tankyevych, O., Talbot, H., and Dokládal, P. (2008). Curvilinear morpho-hessian filter. In *proceedings of the International Symposium on Biomedical Imaging (ISBI)*, pages 1011–1014.

Tankyevych, O., Talbot, H., Dokladal, P., and Passat, N. (2009). Spatially-variant morpho-hessian filter: efficient implementation and application. In *Mathematical Morphology and Its Application to Signal and Image Processing, Proceedings of the 9th International Symposium on Mathematical Morphology (ISMM) 2009*, pages 137–148, Groningen, The Netherlands.

Truc, P., Khan, M., Lee, Y., Lee, S., and Kim, T. (2009). Vessel enhancement filter using directional filter bank. *Computer Vision and Image Understanding*, 113(1):101–112.

Tsitsiklis, J. N. (1995). Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538.

Tuzikov, A., Soille, P., Jeulin, D., Bruneel, H., and Vermeulen, M. (1992). Extraction of grid lines on stamped metal pieces using mathematical morphology. In *Proc. 11th IAPR International Conference on Pattern Recogni tion, Conference A: Computer Vision and Applications*, volume 1, pages 425–428, The Hague.

Unger, M. (2012). *Convex Optimization for Image Segmentation.* PhD thesis, T. U. Graz. http://www.icg.tugraz.at/publications/www.icg.tugraz.at/publications/pdf/convex-optimization-for-image-segmentation.

Unger, M., Pock, T., and Bischof, H. (2008). Interactive globally optimal image segmentation. Technical Report ICG–TR–08/02, Institute for Computer Graphics and Vision, Graz University of Technology.

Unser, M. and Van De Ville, D. (2009). Higher-order Riesz transforms and steerable wavelet frames. In *Proceedings of the 2009 IEEE International Conference on Image Processing (ICIP'09)*, pages 3801–3804, Cairo, Egypt.

Urbach, E. and Wilkinson, M. (2008). Efficient 2-d grayscale morphological transformations with arbitrary flat structuring elements. *Image Processing, IEEE Transactions on*, 17(1):1–8.

Van Droogenbroeck, M. and Buckley, M. (2005). Morphological erosions and openings: fast algorithms based on anchors. *Journal of Mathematical Imaging and Vision*, 22(2):121–142.

Van Ginkel, M., Van de Weijer, J., Van Vliet, L., and Verbeek, P. (1999). Curvature estimation from orientation fields. In *PROCEEDINGS OF THE SCANDINAVIAN CONFERENCE ON IMAGE ANALYSIS*, volume 2, pages 545–552.

van Ginkel, M., van Vliet, L., and Verbeek, P. (1996-2001). Applications of image analysis in orientation space. In Vossepoel, A. and Vos, F., editors, *Fourth Quinquennial Review*, NVPHBV, pages 355–370, Delft. Dutch Society for Pattern Recognition and Image Processing.

van Herk, M. (1992). A fast algorithm for local minimum and maximum filters on rectangular and octogonal kernels. *Pattern Recognition Letters*, 13:517–521.

van Vliet, L. and Verbeek, P. (1995). Segmentation of overlapping objects. In *Abstracts of the ASCI Imaging Workshop*, pages 5–6.

Vandroogenbroeck, M. and Talbot, H. (1996). Fast computation of morphological operations with arbitrary structuring elements. *Pattern Recognition Letters*, 17:1451–1460.

Veksler, O. (1999). *Efficient graph-based energy minimization.* PhD thesis, Cornell University.

Veksler, O. (2007). Graph cut based optimization for MRFs with truncated convex priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN.

Veksler, O. (2009). Multi-label moves for MRFs with truncated convex priors. In *The 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition EMMCVPR'09*, pages 1–8.

Viero, T. and Jeulin, D. (1995). Morphological extraction of line networks from noisy low-constrast images. *Journal of Visual Communication and Image Representation*, 6(4):333–347.

Villegas, O., de Jesus Ochoa Dominguez, H., and Sánchez, V. (2008). A comparison of the bandelet, wavelet and contourlet transforms for image denoising. In *Artificial Intelligence, 2008. MICAI'08. Seventh Mexican International Conference on*, pages 207–212. IEEE.

Vincent, L. (1998). Minimal path algorithms for the robust detection of linear features in images. In *Mathematical Morphology and its Applications to Image and Signal processing*, volume 12 of *Computational Imaging and Vision*, pages 331–338, Amsterdam. Kluwer. Proceedings for ISMM'98.

Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598.

Vogel, C. R. and Oman, M. E. (1996). Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238.

Weickert, J. (1998). *Anisotropic diffusion in image processing*. B.G. Teubner, Stuttgart.

Weickert, J., Romeny, B., and Viergever, M. (2002). Efficient and reliable schemes for nonlinear diffusion filtering. *Image Processing, IEEE Transactions on*, 7(3):398–410.

Witkin, A. P. (1983). Scale-space filtering. In *Proc. 8th Int. Joint Conf. Art. Intell*, pages 1019–1022, Karlsruhe, Germany.

Woodford, O. J., Torr, P. H. S., Reid, I. D., and Fitzgibbon, A. W. (2008). Global stereo reconstruction under second order smoothness priors. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Xiao, J. and Shah, M. (2007). Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1644–1659.

Zach, C., Gallup, D., Frahm, J.-M., and Niethammer, M. (2008). Fast global labeling for real-time stereo using multiple plane sweeps. In *VMV*, pages 243–252.

Zhang, B., Fadili, J., and Starck, J. (2008). Wavelets, ridgelets, and curvelets for poisson noise removal. *Image Processing, IEEE Transactions on*, 17(7):1093–1108.

Zhu, S. and Yuille, A. (2002). Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(9):884–900.

Index

aneurisms, 22

bandelets, 46
benchmark, 23
blood vessels, 22
Bresenham lines, 50

CFA, 24, 30
closing, 49
codimension, 28
colour filter array, 24
covector, 33, 37
crack, 28
curvelet transform, 45

DCTV, 153
demosaicing, 23
directional filter bank, 41
directional filter banks, 40

filter banks, 39
fingerprints, 41
finite impulse response, 43

Gabor filters, 40
gradient, 33

Hessian, 34

image analysis, 16
Image deconvolution, 156

markers, 16
metrication artifact, 131
move
    swap, 120

Neurites, 20

occlusion, 19
opening, 49
orientation space, 41

Parallel ProXimal Algorithm (PPXA), 155
power watershed, 122
Projection operator, 155
pyramid, 34, 39

retina, 21
rotated filters, 40

scale, 19
scale space, 41
scale-space, 34
scene, 16
seeds, 16
segmentation, 15
    bottom-up, 15
    top-down, 15
steerable filters, 37
stent, 22
structure tensor, 37

textures, 22
Translation invariance, 55
translation invariance, 57
Tukey biweight, 122

vesselness, 34

wavelets, 41
Weight function, 154