Depth map estimation from stereo images & mathematical morphology

Jean-Charles Bricola

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à MINES ParisTech

Depth map estimation from stereo images & mathematical morphology

Estimation de cartes de profondeur à partir d'images stéréo et morphologie mathématique

Ecole doctorale n°432

SCIENCES DES MÉTIERS DE L'INGÉNIEUR

Spécialité MORPHOLOGIE MATHÉMATIQUE

COMPOSITION DU JURY :

Mme. Sabine SÜSSTRUNK
EPFL • Ecole Polytechnique Fédérale de Lausanne, Présidente

M. Peter DE WITH
TUE • Technische Universiteit Eindhoven, Rapporteur

M. François BREMOND
INRIA • Institut National de la Recherche en Informatique et en Automatique, Rapporteur

M. Klaas Jan DAMSTRA
Grass Valley, Membre du jury

M. Michel BILODEAU
MINES ParisTech, Membre du jury

M. Serge BEUCHER
MINES ParisTech, Membre du jury

Soutenue par
JEAN-CHARLES **BRICOLA**

le 19 octobre 2016

Dirigée par **Serge BEUCHER**
& **Michel BILODEAU**

MINES ParisTech | PSL RESEARCH UNIVERSITY PARIS

Depth map estimation from stereo images & mathematical morphology

Jean-Charles Bricola

Contents

Introduction

The problem of computing a depth map from a pair of stereo images is a classic one in the field of computer imaging science. Broadly speaking, it is essentially a matter of finding corresponding pixels in two separate images. Although there have been considerable advances in this area since the 1970s, recent availability of databases in which images are either devoid of texture, or are subject to various kinds of defects such as noise or blurriness, has accentuated the need for new approaches, which place greater emphasis on image pre-processing and for which image segmentation is brought into play.

In most cases, the segmentation itself, in relation to computing depth maps, consists of partitioning the images into a set of homogeneous connected components with the aim of highlighting the potential objects or parts of objects, which compose the scene. Using such segments within the "matching process" which yields the final depth map, enables algorithms to take into account their shape or morphology. Furthermore, the segmentation allows some parts of the ambiguities induced by the occlusion phenomenon to be more easily solved since regions seldom undergo total occlusions but more often semi-occlusions with respect to one image of the stereo pair. Finally, regions provide pertinent information about the pieces in which we should expect the depth function to evolve without discontinuities until a high level of a segmentation hierarchy is attained.

Using regions for depth computation is however *not* straightforward and raises some difficulties, the first of which stems from the matching itself. When considering partitions computed independently for both stereo images, it is not necessarily the case that a one-to-one correspondence exists between the regions. Working with strictly equivalent partitions is therefore beneficial and this thesis will provide a method for their generation. Nonetheless, as opposed to point-based approaches, the pairing of regions alone is insufficient to yield a depth map. It is usually necessary to find the transformation which best superimposes two similar regions. Then, in order to use a computationally tractable transformation model, several questions need to be answered. Which level(s) of a segmentation hierarchy should be used? Is it sufficient to base the registration of two regions on their shape only? What happens if the segmentation fails to distinguish two separate objects in a scene?

The purpose of this thesis is to provide answers to these questions, and ultimately to show the benefits of region-based approaches to the computation of depth maps with respect to two very different and challenging scenarios. The first is the generation of depth maps from stereo imagery, in which some areas are out of focus and which is captured by two relatively close lenses. The second is the application of one of our methods to the new Middlebury 2014 database which provides wide-baseline stereo imagery with a great many homogeneous and textured areas comprising both thin and large objects.

An overview of the thesis contents

The thesis is divided into two parts. The first can be viewed as an introduction to depth map estimation and mathematical morphology, which will prove useful when approaching the second part. Beyond its introductory aspect, part one details and illustrates key observations on which our approaches are based.

Stereo image formation The opening chapter is a reminder of the geometrical relationships which manifest within a pair of stereo images. We shall, in the rest of this text, often discuss and be interested in the *disparity*, which measures the displacement of a point across two images. In fact, disparity and depth are closely related, as will be explained, and simple mathematical relationships exist between them when the images are *rectified*, which will always be assumed to be the case in our study.

The problem of depth map computation Once the stereo pair has been rectified, the computation of a depth map can take place. The second chapter highlights two sides of the problem. First, the *measurement* of disparities: which features are good and which are misleading when trying to establish correspondences, and what impact does the segmentation scale have on the establishment of correspondences. Assigning disparity values to parts of an image is equivalent to warping these parts with the other image of the stereo pair. Unfortunately, because of the occlusion phenomenon, a warp based solely on the image content is impracticable. Hence the second side of the problem: *estimation*.

Mathematical Morphology The final chapter of this introductory part presents the morphological operators exploited in this work. We explain how such operators enable the construction of powerful image filters and segmentation tools. Naturally, a special focus is set on the watershed transformation controlled by markers. This transformation is utilised at various levels of our work, in particular for the generation of segmentations suitable for stereo analysis, for the co-segmentation of stereo images, and for the disparity map interpolation based on distance functions. This final introductory chapter concludes with the generalised and geodesic distance functions, which enable interpolations in more complex scenarios.

The second and main part of this text is devoted to the thorough presentation of our region-based approaches. Algorithms and results, including intermediate results, are provided throughout the dissertation.

Segmentation Segments can be considered good for stereo matching if they preserve some regularity and if they partition the scene into salient regions. In order to segment a scene based on criteria such as area and contrast while using the watershed transformation, it is first necessary to compute an appropriate topographical surface which highlights the contours in the scene with an intensity matching human perception. The enhanced regularised gradient presented in this chapter satisfies these requirements. Then, following a survey of some morphological segmentations exploiting area and contrast criteria, we present our own adaptive over-segmentation algorithm whereby each region of the partition has an area above a minimum threshold; a threshold determined by a function of the region saliency in the scene.

Superimpositions Once the partitions have been computed, one can proceed to find for each segment a transformation which yields an optimal superimposition with its analogous segment in the other image of the stereo pair. Two different approaches to the matter are presented. First, we consider the translational transformations producing *regional disparities*. We explain how to compute these from image gradients and lightness, and which geometrical assumptions need to be fulfilled. Second, we consider local pairings which take into account the shape of the regions. We show how to extract reliable and sufficiently dense disparity data from these local matches, based on multi-scale analyses and morphological filtering. At the end of the chapter, we perform a comparison of these two alternatives.

Co-Segmentations Regional disparities may lack in precision while the disparity maps produced by our local matching algorithm leave some pixels without a disparity measure. However, it is possible from such data to generate equivalent segmentations of a pair of stereo images. This chapter provides full details for the generation of these *co-segmentations*, with a particular focus on how semi and total occlusions are handled by the proposed algorithm. Co-segmentations are important because they provide strong indications about how to further constrain the stereo pairings, and thus they are used within one of our estimation algorithms.

Estimation Since no disparity can be measured across occluded areas, completing depth maps involves some estimation procedure. Three main estimation techniques are presented in this chapter, each illustrated by a particular application with respect to stereo matching. The first technique is based on a linear estimator called “kriging” and exploits very sparse but accurate disparity data. We discuss an application where the co-segmentations are used to compute contour disparities and where kriging analyses these disparities to interpolate the disparity map. When the data is denser, such as that obtained using the local matches computed across multiple scales, an interpolation based on distance functions operates effectively to fill the holes of the

disparity map. Finally, we present an estimator based on the maximum a posteriori inference. This last proves particularly useful on regional disparity maps and essentially involves the correction of erroneous disparity measures.

Part I

Observations

Résumé du chapitre 1

Dans ce chapitre, nous rappelons les relations géométriques caractérisant deux images stéréoscopiques. Ces relations permettent de déduire la profondeur d'un point de la scène à partir de ses projetés dans les deux images stéréoscopiques, ou encore de déduire la fonction de profondeur associée à un plan de l'espace 3D, étant donné la transformation qui lie ses projections dans les deux images constituant la paire stéréo.

Les algorithmes de calcul de cartes de profondeur présentés dans cette thèse partent du principe que les deux images stéréoscopiques sont rectifiées. Autrement dit, les points de la scène se projettent dans les deux images stéréo avec des ordonnées identiques mais des abscisses différentes. La différence d'abscisse entre les deux projetés d'un même point 3D correspond à la disparité, et est inversement proportionnelle à la profondeur de ce point 3D. Ainsi, lorsque nous calculons une carte de profondeur par rapport à l'une des deux images stéréoscopiques, il est nécessaire de trouver, pour chaque point de l'image considérée, sa correspondance dans l'autre image de la paire stéréo. Or, c'est bien là tout le problème du calcul de cartes de profondeur. A ce titre, le chapitre 2 permettra au lecteur de bien cerner les difficultés liées à l'analyse d'images stéréoscopiques et d'apprécier la mesure dans laquelle les solutions existantes résolvent ces problèmes.

Chapter 1

Geometry of Stereo Images

A pair of stereoscopic images is composed of two images, each representing a different viewpoint of a real-world scene. The purpose of this chapter is to review the geometrical relationships which exist between stereo images, and which are important in the context of depth map computation. Section 1.1 demonstrates the mathematical relations allowing one to recover depth information related to a point or a plane of the scene, given their projections onto the images of the stereo pair. Section 1.2 shows how the rectified configuration simplifies most of these relations.

1.1 General relations

In this work, we assume that any image is the product of a *pinhole camera*. A pinhole camera is characterised by two distinct properties. The coordinates of the camera centre, as well as its orientation define the extrinsic parameters of the camera. Both are relative to the world's origin and axes orientation. In addition, the camera is virtually associated with an image plane onto which a point of the scene projects. That image plane is determined by the intrinsic parameters of the camera, comprising the principal point, defined as the intersection of the image plane with the optical ray originating from the camera centre, and the focal length, which equals the distance separating the camera centre from the principal point.

Let P be an arbitrary point of the scene. We call $\mathbf{X}_1 = (X_1, Y_1, Z_1)^\intercal$ its coordinates *relative* to the centre and orientation axes of camera C_1. The coordinates (x_1, y_1) of the image plane associated with camera C_1 onto which P projects, are defined according to the intercept theorem, as illustrated in figure 1.1. In homogeneous coordinates, this transformation is driven by the intrinsic parameters of camera C_1 and is expressed as:

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \mathbf{K}\mathbf{X}_1 \tag{1.1}$$

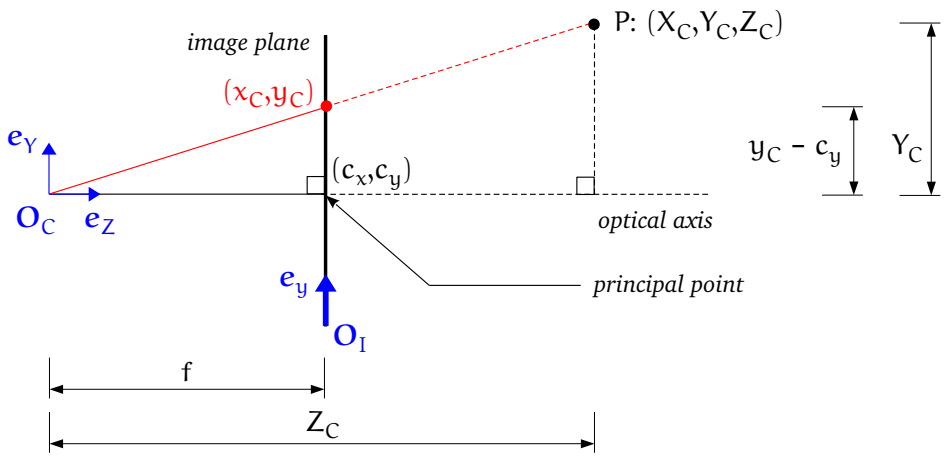


FIGURE 1.1: The projection model of a pinhole camera. The coordinate system of camera C is defined in function of its origin $\mathbf{O}_C$ and its orientation axes $\{\boldsymbol{e}_X, \boldsymbol{e}_Y, \boldsymbol{e}_Z\}$. Within this referential, point P has coordinates (X_C, Y_C, Z_C). It projects onto the image plane with coordinates (x_C, y_C) relative to the image plane origin $\mathbf{O}_I$ and orientation axes $\{\boldsymbol{e}_x, \boldsymbol{e}_y\}$. The internal parameters of the camera are the focal length f and the principal point (c_x, c_y). According to the intercept theorem, we have $(y_C - c_y)/f = Y_C/Z_C$ and a similar relation holds between x_C, c_x, X_C and Z_C. Note that in order to simplify the visualisation of the perspective projection, the X-coordinate has been discarded from the illustration.

where f corresponds to the focal length and (c_x, c_y) represents the principal point of the image plane related to camera C_1. The image coordinates are retrieved from the homogeneous coordinates as follows: $x_1 = \tilde{x}_1/\tilde{z}_1$ and $y_1 = \tilde{y}_1/\tilde{z}_1$.

1.1.1 Depth from the projections of a 3D point

In the problem of depth map computation, we seek the depth Z_1 of the point having (x_1, y_1) as projection. Let us now look at how this coordinate Z_1 can be recovered using a pair of stereo images. The extrinsic parameters of a camera enable us to determine the manner in which the coordinates of a point in the 3D world, map to its equivalent coordinates in respect to the reference frame of that camera. These parameters are typically characterised by the composition of a rotation and a translation. Let $\mathbf{X} = (X, Y, Z)^\intercal$ be the world coordinates of point P. $\mathbf{R}_1$ and $\mathbf{t}_1$ are respectively the rotation matrix and translation vector used to transform $\mathbf{X}$ into $\mathbf{X}_1 = [\mathbf{R}_1 \mid \mathbf{t}_1]\mathbf{X}$. The corresponding transformation in homogeneous coordinates is expressed by:

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} r_{1,1}^{(1)} & r_{1,2}^{(1)} & r_{1,3}^{(1)} & t_x^{(1)} \\ r_{2,1}^{(1)} & r_{2,2}^{(1)} & r_{2,3}^{(1)} & t_y^{(1)} \\ r_{3,1}^{(1)} & r_{3,2}^{(1)} & r_{3,3}^{(1)} & t_z^{(1)} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where the $r_{i,j}^{(1)}$ are the entries of $\mathbf{R}_1$ for $i, j \in \{1, 2, 3\}$, and $\mathbf{t}_1 = (t_x^{(1)}, t_y^{(1)}, t_z^{(1)})^\intercal$. We introduce a new camera C_2 to the scene. For the sake of simplicity, we assume that this camera has the same intrinsic parameters as camera C_1 and we let $\mathbf{X}_2 = [\mathbf{R}_2 \mid \mathbf{t}_2]\mathbf{X}$ be the coordinates of point P with respect to C_2. It is easy to derive the rigid transformation $[\mathbf{R} \mid \mathbf{t}]$ which maps $\mathbf{X}_1$ to $\mathbf{X}_2$, since $\mathbf{X}_2 = [\mathbf{R}_2 \mid \mathbf{t}_2] \left([\mathbf{R}_1 \mid \mathbf{t}_1]^{-1} \mathbf{X}_1 \right)$.

Given the projections of P onto the image planes of C_1 and C_2, as well as the aforementioned calibration parameters, the depth of P can be computed as follows. First, let $x_k = f x'_k + c_x$ and $y_k = f y'_k + c_y$ for any $k \in \{1, 2\}$. Because of equation 1.1, $x'_k = X_k / Z_k$ and $y'_k = Y_k / Z_k$. Since we know how to relate $\mathbf{X}_2$ to $\mathbf{X}_1$, the former two equations can be developed:

$$\begin{aligned} x'_2 &= \frac{X_2}{Z_2} \\ &= \frac{(r_{1,1}X_1 + r_{1,2}Y_1 + r_{1,3}Z_1) + t_x}{(r_{3,1}X_1 + r_{3,2}Y_1 + r_{3,3}Z_1) + t_z} \\ &= \frac{Z_1 (\overbrace{r_{1,1}x'_1 + r_{1,2}y'_1 + r_{1,3}}^{E_1}) + t_x}{Z_1 (\underbrace{r_{3,1}x'_1 + r_{3,2}y'_1 + r_{3,3}}_{E_3}) + t_z} \end{aligned} \tag{1.2}$$

where $r_{i,j}$ are the coefficients of the rotation matrix $\mathbf{R}$ for $i, j \in \{1, 2, 3\}$, and $\mathbf{t} = (t_x, t_y, t_z)^\intercal$. We obtain a similar expression for y'_2.

$$y'_2 = \frac{Z_1 (\overbrace{r_{2,1}x'_1 + r_{2,2}y'_1 + r_{2,3}}^{E_2}) + t_y}{Z_1 (\underbrace{r_{3,1}x'_1 + r_{3,2}y'_1 + r_{3,3}}_{E_3}) + t_z} \tag{1.3}$$

From relations 1.2 and 1.3, we deduce that Z_1 is not defined when $\mathbf{t} = \mathbf{0}$. In other words, in order to perform depth estimation from a pair of stereo images, it is essential that the two camera centres should not be the same. Given that E_1, E_2 and E_3 depend solely on the calibration parameters and the coordinates (x_1, y_1), Z_1 can be expressed as a function of the image coordinates of the two projections:

$$\begin{aligned} x'_2(Z_1 E_3 + t_z) &= Z_1 E_1 + t_x \\ Z_1(x'_2 E_3 - E_1) &= t_x - x'_2 t_z \\ Z_1 &= \frac{t_x - x'_2 t_z}{x'_2 E_3 - E_1} \end{aligned} \tag{1.4}$$

And similarly by developing from y'_2, we obtain that:

$$Z_1 = \frac{t_y - y'_2 t_z}{y'_2 E_3 - E_2} \tag{1.5}$$

Either equation 1.4 or 1.5 can be chosen to compute Z_1. The preferred choice will depend on whether the denominator appearing in each relation, is different from zero. Recall that Z_1 refers to the depth of the 3D point of interest, with respect to the coordinate system of camera C_1. When the coordinate systems of C_1 and the scene are the same, Z_1 will correspond to the depth coordinate of the point in the scene.

1.1.2 Depth from the projections of a 3D plane

So far, we have learnt how to relate the depth of a 3D point to the coordinates of its projections onto each of the stereo images. We are now interested in computing a depth function associated with a plane lying in front of the camera's objective.

For the rest of this paragraph, we assume that every coordinate is expressed in the referential of camera C_1. Let π be a plane in 3D space defined by equation $aX + bY + cZ + k = 0$ and consider the point of coordinates $\mathbf{x}_1 = (x, y, f)^\intercal$ belonging to the image plane of C_1. By tracing a ray from the optical centre of C_1 travelling past $\mathbf{x}_1$, it is possible to find an intersection with π, say $\mathbf{X}_1 = (X_1, Y_1, Z_1)^\intercal$ using the following parameterisation: $X_1(t) = xt$, $Y_1(t) = yt$ and $Z_1(t) = ft$. The time at which this ray going past $\mathbf{x}_1$ intersects π, is given by equation 1.6.

$$t(\mathbf{x}_1, \pi) = \frac{-k}{ax + by + cf} \tag{1.6}$$

Let us call $\boldsymbol{\pi} = \frac{1}{k}(a, b, c)^\intercal$ the plane coordinates. Equation 1.6 reduces to $t(\mathbf{x}_1, \pi) = -(\boldsymbol{\pi}^\intercal \mathbf{x}_1)^{-1}$. The relation between $\mathbf{X}_1$ and $\mathbf{X}_2$, denoting the coordinates of P with reference to the coordinate systems of cameras C_1 and C_2 respectively, can now be further developed.

$$\begin{aligned} \mathbf{X}_2 &= \mathbf{R}\mathbf{X}_1 + \mathbf{t} \\ &= t(\mathbf{x}_1, \pi)\mathbf{R}\mathbf{x}_1 + \mathbf{t} \\ &= \mathbf{R}\mathbf{x}_1 - (\boldsymbol{\pi}^\intercal \mathbf{x}_1)\mathbf{t} \\ &= (\mathbf{R} - \mathbf{t}\boldsymbol{\pi}^\intercal)\mathbf{x}_1 \end{aligned}$$

We observe that $\mathbf{x}_1$ represents the de-calibrated homogeneous coordinates of the projection of $\mathbf{X}_1$ into the view associated with C_1. Therefore $\mathbf{x}_1 = \mathbf{K}^{-1}\tilde{\mathbf{x}}_1$, where $\tilde{\mathbf{x}}_1$ corresponds to the homogeneous coordinates of the pixel representing $\mathbf{X}_1$ in the image plane of C_1. Similarly, the homogeneous coordinates of the pixel corresponding to $\mathbf{X}_2$ in the view associated with C_2, are given by $\tilde{\mathbf{x}}_2 = \mathbf{K}\mathbf{X}_2$. We therefore deduce that:

$$\tilde{\mathbf{x}}_2 = \mathbf{K}(\mathbf{R} - \mathbf{t}\boldsymbol{\pi}^\intercal)\mathbf{K}^{-1}\tilde{\mathbf{x}}_1 \tag{1.7}$$

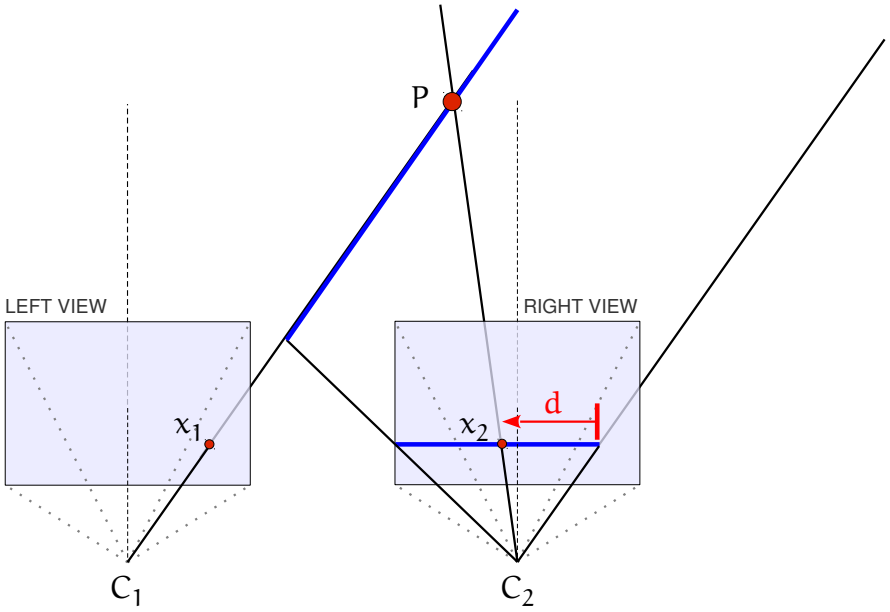


FIGURE 1.2: Relation between depth and disparity for a pair of rectified stereo images. Point P projects onto the left and right views of the stereo pair with the same ordinate but different abscissa. The quantity $x_1 - x_2$ corresponds to the *disparity* and is inversely proportional to the depth being searched for. The further away P is situated from the camera centres, the smaller is the disparity. We see that if $x_1 = x_2 \Leftrightarrow d = 0$, then the rays originating from the camera centres C_1 and C_2, going past x_1 and x_2 respectively, meet at infinity.

The linear operator $\mathbf{H} = \mathbf{K}(\mathbf{R} - \mathbf{t}\boldsymbol{\pi}^\intercal)\mathbf{K}^{-1}$ transforming homogeneous coordinates $\tilde{\mathbf{x}}_1$ into $\tilde{\mathbf{x}}_2$ is a planar homography, represented as a 3×3 matrix. Therefore, the transformation which warps the projections of a planar object from one view to another is the homography $\mathbf{H}$. If we are able to estimate the latter, the plane equation can be recovered from relation $\boldsymbol{\pi}^\intercal = \frac{1}{3}(\mathbf{t}^{-1})^\intercal(\mathbf{R} - \mathbf{K}^{-1}\mathbf{H}\mathbf{K})$ and therefore the depth coordinates of all the points belonging to the plane can be deduced.

1.2 Relations specific to the rectified configuration

Stereo images are said to be *rectified* when the orientation of the camera axes has been preserved, i.e. when $\mathbf{R}$ is the identity matrix, and when the ordinates of the corresponding points are aligned, requiring that $y_1 = y_2$, which can be achieved for $t_y = t_z = 0$, according to equation 1.3. For that particular choice of parameters, the ratio in equation 1.5 is undefined. Therefore, the depth must be recovered using equation 1.4. We have $E_3 = r_{3,3} = 1$, as well as $E_1 = r_{1,1}x'_1 = x'_1$. Therefore,

$$Z_1 = -\frac{t_x}{x'_1 - x'_2}$$

Recalling that $x'_k = (x_k - c_x)/f$, for $k \in \{1, 2\}$, the depth is ultimately given by equation 1.8.

$$Z_1 = -f\frac{t_x}{x_1 - x_2} \tag{1.8}$$

The quantity $x_1 - x_2$ corresponds to what is called *the disparity*, and is, according to equation 1.8, inversely proportional to the depth being searched for. For the sake of completeness, note that when C_1 and C_2 respectively capture the left and right views of a scene, $x_1 - x_2 > 0$, but $t_x < 0$. Therefore, $Z_1 > 0$.

Relation between 3D planes and disparity functions

We can rewrite equation 1.8 more compactly, as:

$$d(\mathbf{x}_1) = -f \cdot \frac{t_x}{Z_1(\mathbf{x}_1)}$$

so that $d(\mathbf{x}_1)$ and $Z_1(\mathbf{x}_1)$ denote respectively the disparity and the depth associated with point $\mathbf{x}_1 = (x, y, f)^\intercal$, belonging to the image plane of camera C_1. If we now reconsider the scenario of section 1.1.2, where $Z_1(\mathbf{x}_1) = f \cdot t(\mathbf{x}_1, \pi)$ because $\mathbf{x}_1$ is the projection of a point belonging to the 3D plane π, then:

$$d(\mathbf{x}_1) = -\frac{t_x}{t(\mathbf{x}_1, \pi)} = \frac{t_x}{k}(ax + by + c) \tag{1.9}$$

This result is particularly useful, as it shows that the disparity function can be expressed as a plane equation within any portion of the image domain segmenting an object, which is a plane in the 3D scene. Of course, the validity of this assertion is contingent upon the stereo images having been rectified.

Image transformation induced by planes

In the rectified configuration, the homography allowing the warp of the projections of a plane onto the stereo images, can also be arranged as a much simpler transformation. Setting $\mathbf{R}$ to the identity matrix and $t_y = t_z = 0$, one obtains:

$$\mathbf{I} - \mathbf{t}\boldsymbol{\pi}^\intercal = \left[\begin{array}{cc|c} 1 - t_x\frac{a}{k} & -t_x\frac{b}{k} & -t_x\frac{c}{k} \\ 0 & 1 & 0 \\ \hline 0 & 0 & 1 \end{array}\right]$$

which is therefore an *affine* transformation. Given that the intrinsic calibration matrix $\mathbf{K}$ is defined such that $\mathbf{K}_{3,1} = \mathbf{K}_{3,2} = 0$, we deduce that the homography $\mathbf{H}$ expressed by equation 1.7 becomes an affine transformation under the condition that the stereo images are rectified.

Further reading

The process of rectifying an arbitrary pair of stereo images is beyond the scope of this text. It should be noted that the solution of the rectification problem in stereo is not unique. We refer the reader to [Ayache and Hansen, 1988], which is one of the pioneer works on image rectification, as well as to [Loop and Zhang, 1999], where more emphasis is laid on the minimisation of distortions. The reader can find an additional source of information about stereo image formation and interpretation in [Horaud and Monga, 1995], as well as a careful treatment of two-views geometry in [Hartley and Zisserman, 2004].

Summary

In this chapter, we have presented and proved the geometrical relationships characterising stereo images. Such relationships are useful when relating the correspondences of two image pixels to the depth of the 3D point they represent, or when relating the transformation which maps the projections of a 3D plane to its actual depth function.

The algorithms of depth map computation presented in the rest of this thesis, consider the rectified configuration. This allows us to view the disparity between two corresponding points as a measure being inversely proportional to the sought depth. The real problem of depth map computation is however to establish correspondences between the pixels of the two stereo images. The following chapter reviews many of the methods developed to date, and highlights the main problems relating to the computation of disparities.

Résumé du chapitre 2

Ce chapitre passe en revue les difficultés rencontrées dans le cadre de la recherche de mises en correspondance entre les deux images stéréoscopiques.

En calculant une image des différences absolues entre les fonctions d'intensité associées à la vue de gauche et à la vue de droite, soit une image contenant des coûts de superposition entre les deux images stéréo, nous remarquons qu'une ombre de superposition balaye le plan de l'image de superposition au fur et à mesure que nous décalons la vue de droite vers le bord droit du plan de l'image. Le décalage pour lequel un pixel de la vue de gauche est recouvert par l'ombre de superposition correspond à sa disparité réelle.

Afin de mettre cette ombre de superposition en évidence, il peut être utile d'agréger les coûts de superposition de pixels voisins, et ce pour chaque décalage possible. Dans ce cas, pour que les agrégations de coûts soient pertinentes, les segmentations des deux images stéréo doivent être prises en compte. En effet, à chaque coût de superposition correspond deux régions: l'une provenant de la vue de gauche, et l'autre provenant de la vue de droite. Par souci de cohérence, seules les agrégations de coûts provenant de la même intersection régionale peuvent être tolérées. De plus, afin de tenir compte des régions représentant des objets inclinés par rapport au plan d'image de la caméra, l'agrégation de coûts entre images de superpositions obtenues pour différents décalages s'avère également essentielle. C'est la raison pour laquelle nous présentons les volumes de superpositions d'images (DSV), desquels nous espérons extraire les hyperplans représentatifs des ombres de superposition, sur une base régionale, et au travers de plusieurs plans de disparités.

Il faut bien retenir que le calcul de cartes de profondeur est un problème inverse. Cela signifie que, pour une image donnée, il existe des pixels n'ayant pas de correspondance dans l'autre image de la paire stéréo. Ces pixels sont dits « occultés », car ils correspondent à des points de la scène qui ne se projettent pas dans l'autre image de la paire stéréoscopique. Il nous faut donc un modèle d'estimation adéquat pour attribuer à ces pixels, des valeurs de disparités plausibles. Un bon modèle d'estimation doit également servir à résoudre les ambiguïtés de mises en correspondance au niveau des zones de l'image qui sont homogènes. Nous avons montré qu'il est toujours préférable de se fier à des informations de profondeur internes aux régions de

l'image. Lorsque cela n'est pas possible, les disparités de contour peuvent être utilisées en dernier recours, après avoir déterminé à quelles régions ces contours appartiennent réellement.

Pour que les agrégations régionales soient cohérentes, il serait idéal que les bordures de régions recouvrent les zones de l'image, là où la disparité réelle subit des discontinuités. Les chapitres 3 et 4 présentent, à ce titre, les outils de segmentation que nous avons déployés. Les chapitres 5, 6 et 7 exploitent les observations rassemblées dans ce chapitre lors de la réalisation de nos méthodes de calcul de cartes de profondeur.

Chapter 2

Stereo Image Analysis

We shall introduce this chapter with a simple example, as illustrated in figure 2.1. Suppose we are given a pair of rectified stereo images such that the corresponding structures preserve the same brightness. We now decide to superimpose the two images, and look at their absolute differences while incrementally shifting the right view horizontally to the right-hand side of the image plane. As the intensity of the shift increases, we notice a shadow progressively sweeping the image plane. In fact, this shadow highlights those areas of the left image which register perfectly with the shifted version of right image. Therefore, the time at which the shadow travels past a pixel is closely connected to its actual disparity. Based on that observation, it is tempting to devise an algorithm capable of tracking this superimposition shadow in order to yield a depth map. However two problems must first be solved. First, the characterisation of this shadow is not trivial and involves observing the phenomenon at an appropriate scale, as will be shown in section 2.1. Second the superimposition of occluded image areas will never be relevant in terms of brightness comparison. Section 2.2 provides a description and an analysis of the occlusion phenomenon while section 2.3 concludes this chapter on the estimation aspect of depth map computation which, due to the unavailability of some correspondences, is essential for the provision of a full depth map.

In the rest of this chapter, we shall employ the notation and structures presented in table 2.1 and assume that we seek a disparity map $\mathfrak{D}$ with respect to the left view of the rectified stereo pair, $\mathbf{I}_l$.

2.1 Measuring disparities: a problem of scale

The pixel constitutes the lowest possible scale at which matches can be observed. For instance, the disparity map could be computed according to equation 2.1 below:

$$\mathfrak{D}\left[x,y\right]^{(\text{pixel})} = \arg\min_{d} \left|\mathbf{I}_l\left[x,y\right] - \mathbf{I}_r\left[x-d,y\right]\right| \tag{2.1}$$

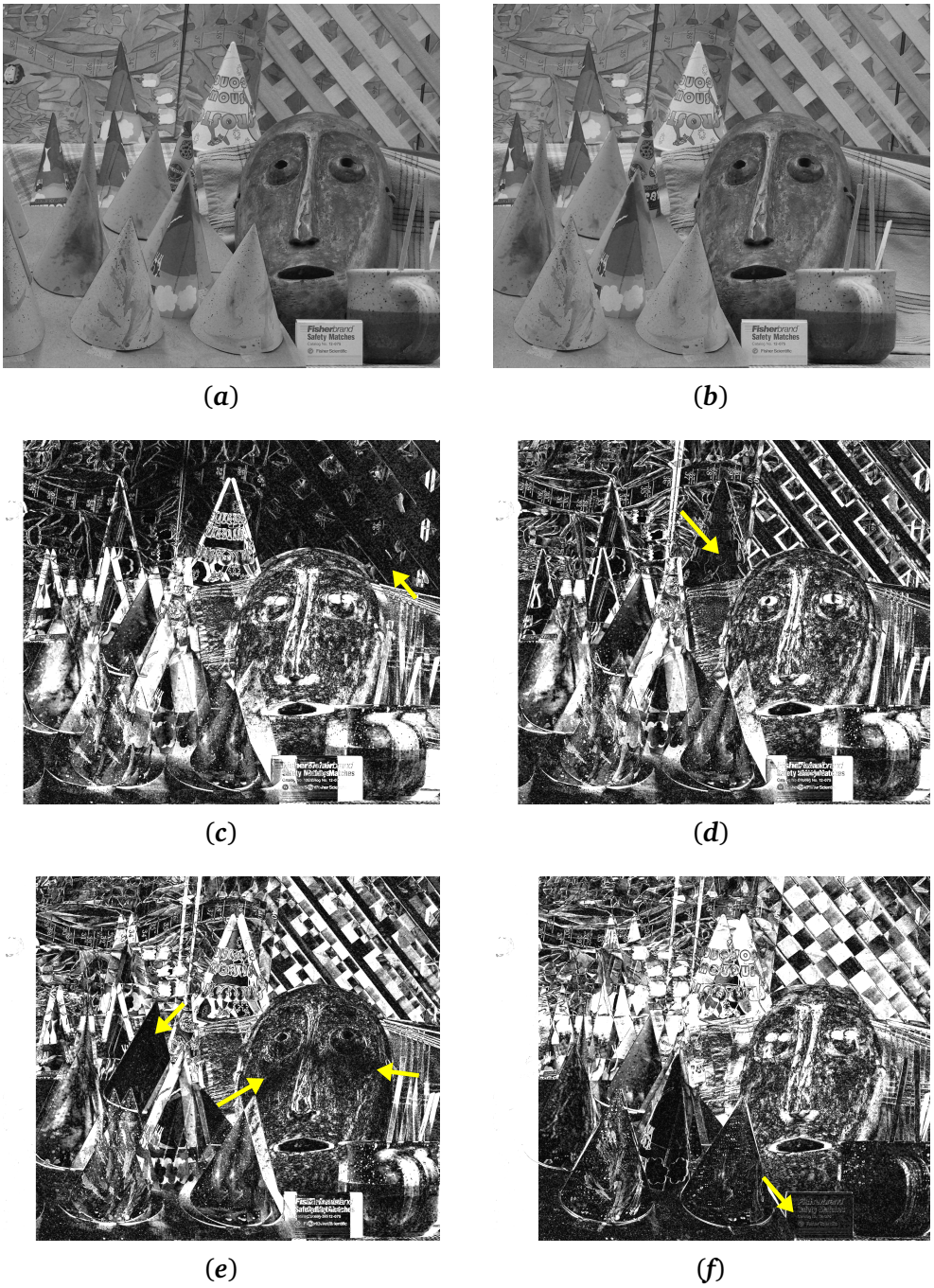


FIGURE 2.1: Stereo analysis and superimpositions. Images (a) and (b) represent the left and right views of the `Cones` scene included in Middlebury 2002 database. Each of the input images has a size of 1800 $\times$ 1500 pixels. The absolute differences between the two images are computed after applying a horizontal shift to the right view towards the right-hand side of the image plane for a magnitude of (c) 84 pixels, (d) 102 pixels, (e) 136 pixels, (f) 186 pixels. For each shift increment, one can observe a superimposition shadow sweeping the areas of the stereo images which correspond. Objects which are almost fronto-parallel to the image plane, like the cones, are entirely swept within a few consecutive horizontal shifts. Tilted objects such as the mask are swept more progressively with a narrower superimposition shadow.

Symbol	Description
Scalars	
x	Pixel abscissa (integer between 0 and image width)
y	Pixel ordinate (integer between 0 and image height)
d	Left-to-right displacement coordinate (integer)
Stereo images	
$\mathbf{I}_l$	Left view of the stereo pair
	$\mathbf{I}_l : (x, y) \mapsto \mathbf{I}_l[x, y]$, the brightness of pixel (x, y) in $\mathbf{I}_l$
$\mathbf{I}_r$	Right view of the stereo pair
	$\mathbf{I}_r : (x, y) \mapsto \mathbf{I}_r[x, y]$, the brightness of pixel (x, y) in $\mathbf{I}_r$
Structures encoding superimpositions	
$\boldsymbol{W}$	Warping Space Volume
	$\boldsymbol{W} : (x_l, x_r, y) \mapsto \boldsymbol{W}[x_l, x_r, y] = \lvert\mathbf{I}_l[x_l, y] - \mathbf{I}_r[x_r, y]\rvert$
$\mathbf{D}$	Disparity Space Volume (DSV)
	$\mathbf{D} : (x, y, d) \mapsto \mathbf{D}[x, y, d] = \boldsymbol{W}[x, x - d, y]$
$\mathfrak{D}$	Disparity map of the left view $\mathbf{I}_l$
	$\mathfrak{D} : (x, y) \mapsto \mathfrak{D}[x, y]$, the disparity allocated to pixel (x, y) in $\mathbf{I}_l$
Aggregation	
$\mathcal{A}(x_i, x_j, y)$	Aggregation support when matching pixels (x_i, y) and (x_j, y)
	$\mathcal{A}(x_i, x_j, y) = \{(x_k, y_k, d_k)\}_{k=1}^{n}$, a set of n points referring to $\mathbf{D}$
Regions	
$\mathbf{R}$	A region of $\mathbf{I}_l$, defined as a set of connected pixels
$\mathbf{R}'$	A region of $\mathbf{I}_r$, defined as a set of connected pixels
$\mathbf{R}'^{(d)}$	A region of $\mathbf{I}_r$, shifted by d pixels towards the right
	$\mathbf{R}'^{(d)} = \{(x, y) \mid (x - d, y) \in \mathbf{R}'\}$

TABLE 2.1: Notation for stereo image analysis

The problem with this equation lies in the fact that there is nothing to prevent very different disparity values from being allocated to two pixels which are simultaneously neighbours and projections of the same object in the scene. The phenomenon is further amplified across regions with constant brightness, which comes as no surprise given that, under those conditions, all pixels would then share almost identical values. For this reason, we will concentrate in this section on the analysis of matches at higher scales. In particular, we shall determine how and to what extent an analysis at a higher scale, i.e. one which considers superimpositions around the pixel of interest and perhaps for different disparities, may solve ambiguities.

2.1.1 Similarity measures

The scale of observation may be increased first by considering patches centred around the candidate pixels for which a match is evaluated. Numerous ways of characterising the similarity or dissimilarity between two patches exist and the we refer the reader to the exhaustive list of measures presented in [Goshtasby, 2012]. We are going to review those which play an important role in stereo and discuss when to use them.

The taxonomy of stereo methods elaborated in [Scharstein and Szeliski, 2002] shows that the sum of squared differences, often referred to as SSD, remains the most popular choice when dealing with stereo imagery acquired under identical illumination conditions. In fact, computing the SSD amounts to squaring the absolute differences of intensities between the superimposed patches and aggregating the results by summation, which is a fairly efficient operation. One can therefore view the SSD as a means of aggregating individual pixel superimposition costs. The same kind of observation would be made for the sum of absolute differences (SAD) and Gaussian convolution.

When image brightness is not preserved across the stereo pair, other measures, which are *not* simply based on individual costs or scores aggregation, are favoured. A typical measure is the normalised cross-correlation, abbreviated as NCC, which exploits the mean and variance of pixel intensities for each patch. Some extensions to that measure, which remain invariant to radiometric changes, have been proposed in [Heo et al., 2008]. The NCC is robust across textured regions due to its locally adaptive normalisation, but is nevertheless insensitive to homogeneous patches of different greylevels. As a result, NCC does not supersede SSD in that respect and should not be used on imagery which benefits from preserved brightness across the stereo pair. Slightly less adaptive than NCC, the intensity ratio-variance may succeed in recognising homogeneous patches for which intensities have been altered, provided that the brightness of equivalent pixels remains identical, up to a scaling factor, which would be assumed to be the case when acquiring images with different exposure times.

Additionaly, alternative measures based on the study of gradient fields have been proposed

in [Scharstein, 1994, Twardowski et al., 2004]. These are convenient when dealing with stereo imagery subject to slight illumination discrepancies. However, on images devoid of texture, their usage requires a proper observation scale and an interpretation of object frontier disparities. In our work, both requirements have been fulfilled by virtue of the use of pertinent image regions, as will be presented in section 5.1. Another image representation of the directional gradients, which stands out as being particularly suitable for computing superimposition costs, is provided by a procedure called the Census transformation [Zabih and Woodfill, 1994]. This transformation maps every pixel of the image plane to a binary code, for which each bit encodes any brightness increase for a particular direction. The pixel-wise superimposition cost is then equal to the Hamming distance between the two corresponding binary sequences. Combined with a simple aggregation strategy (summation for instance), the resulting costs are usually quite discriminative. However, matches based on this transformation are not robust to sensor noise.

The choice of the similarity or dissimilarity measure has a non-negligible impact on the quality of the disparity measurements. However, our work lays more emphasis on the choice of appropriate aggregation supports. Thus far, patches have been assimilated to square windows. This choice is not optimal for two reasons. First, even if the patch centres are correctly superposed, the patch contents may show extreme discrepancies near boundaries of objects with different depths. For that reason, using shape information within the aggregation process is key and some examples will be presented in paragraph 2.1.2. Second, it is unclear how the similarity measure behaves when increasing the window area. This action would be perfectly adequate for a region which is fronto-parallel to the image plane, but the measurements would no longer be valid for tilted regions. The behaviour of tilted regions deserves a thorough study, and is addressed in paragraph 2.1.3.

2.1.2 Shape-adaptive aggregation

State-of-the-art approaches in stereo consider pruning or at least attenuating the impact of wrongly superposed pixels during the aggregation phase. In their work, [De-Maeztu et al., 2012] propose an aggregation scheme based on the use of a geodesic colour distance from the patch centre to the surrounding pixels. As the geodesic distance increases, the aggregation weight assigned to the target pixel of the patch decreases. That is to say only those pixels sharing a colour similar to that of the patch centre and reachable from the patch centre without having to traverse pronounced colour gradients, are likely to be allocated a significant aggregation weight. Segmentation seems therefore well adapted to mimic the procedure just mentioned. In fact, regions have already been used in [Hosni et al., 2010] to constrain the aggregation supports by pruning pixels which do not belong to the patch centre segment. Our work develops from such principles.

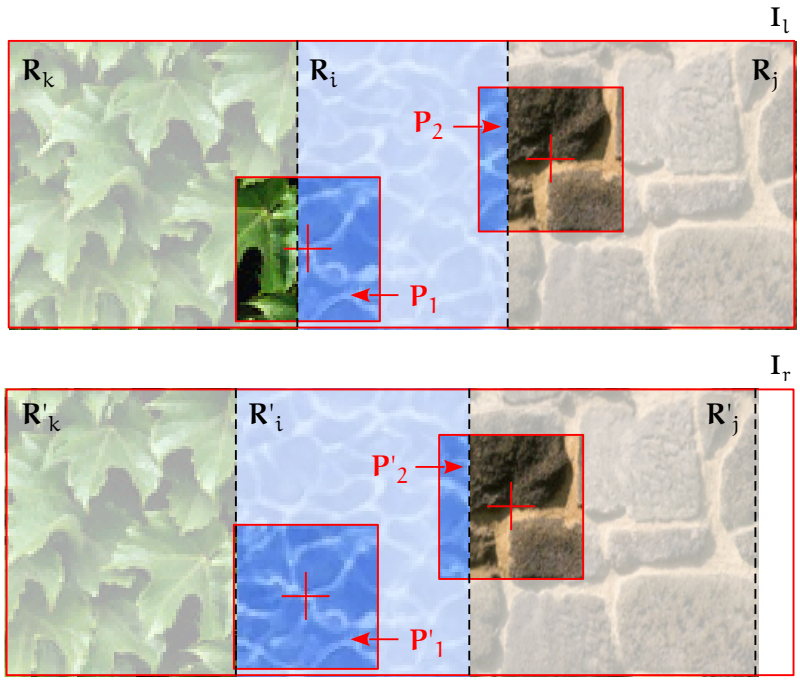


FIGURE 2.2: Scheme illustrating the aggregation consistency problem when patches overlap different objects for the textured case.

An experiment based on a fronto-parallel assumption

Consider the scenario illustrated in figure 2.2. We are provided with two rectified stereo images of the same scene: $\mathbf{I}_l$ and $\mathbf{I}_r$. The red cross enclosed in patch $\mathbf{P}_1$ with respect to the left view matches the red cross enclosed in patch $\mathbf{P}'_1$ in the right view. In a similar way, the red cross enclosed in patch $\mathbf{P}_2$ also corresponds to the red cross enclosed in patch $\mathbf{P}'_2$. For each pair of corresponding pixels between the two stereo images, we can deduce the disparity related to the 3D scene point which projects onto the two corresponding pixels.

Our goal is to determine a mathematical expression of the superimposition cost between any two patches – one extracted from the left view, the other extracted from the right view – in such a manner that the superimposition cost is minimal when the patch centres truly correspond. If we compare the contents of patches $\mathbf{P}_1$ and $\mathbf{P}'_1$, we notice that some parts do not superimpose well, which could result in a non-relevant dissimilarity cost if the entirety of the patches were taken into account. The same observation applies for patches $\mathbf{P}_2$ and $\mathbf{P}'_2$. Consequently, the superimposition costs must depend solely on the patch segments which correctly overlap.

To proceed, we segment the scene into regions $\mathbf{R}_i$, $\mathbf{R}_j$ and $\mathbf{R}_k$ in $\mathbf{I}_l$, and into their corresponding regions in $\mathbf{I}_r$, being $\mathbf{R}'_i$, $\mathbf{R}'_j$ and $\mathbf{R}'_k$ respectively. We assume that:

- Each region represents a particular object of the 3D scene.

- Each region stands *fronto-parallel* to the stereo image planes, meaning that, within a given region, all pixels are the projections of points having the same depth in the 3D scene.
- $\mathbf{R}_i$ is the region lying the farthest away from the camera, while $\mathbf{R}_k$ is the closest to the camera.

We see that patches $\mathbf{P}_1$ and $\mathbf{P}_2$, as well as $\mathbf{P}'_1$ and $\mathbf{P}'_2$ overlap several regions in each view of the stereo pair. In order to determine which areas of the patches are appropriate for deducing that the centres effectively match, we shall concentrate on the intersection between patches and regions. We investigate two configurations: one for which the images are full of non-repetitive texture, and another where the regions are totally homogeneous.

Case 1: Textured regions Let (x_1, y_1) and (x_2, y_2) be the coordinates of the centres of $\mathbf{P}_1$ and $\mathbf{P}_2$ respectively. Suppose that the actual disparity of these centres equals d_1 for patch $\mathbf{P}_1$, and d_2 for patch $\mathbf{P}_2$. From the illustrations of figure 2.2, we can make the following observations:

- $\mathbf{R}_j$ and $\mathbf{R}'_j$ are the regions containing the patch centres of $\mathbf{P}_2$ and $\mathbf{P}'_2$ respectively. We notice that the entirety of $(\mathbf{P}_2 \cap \mathbf{R}_j)$ and $(\mathbf{P}'_2 \cap \mathbf{R}'_j)^{(d_2)}$ are correctly superimposed.
- $\mathbf{R}_i$ and $\mathbf{R}'_i$ are the regions containing the patch centres of $\mathbf{P}_1$ and $\mathbf{P}'_1$ respectively. We notice that only a fraction of $(\mathbf{P}'_1 \cap \mathbf{R}'_i)^{(d_1)}$ is correctly superimposed with $(\mathbf{P}_1 \cap \mathbf{R}_i)$, since $(\mathbf{P}'_1 \cap \mathbf{R}'_i)^{(d_1)}$ covers an area of the image plane, which is larger than that of $(\mathbf{P}_1 \cap \mathbf{R}_i)$. We can explain this phenomenon by the fact that some parts of $(\mathbf{P}'_1 \cap \mathbf{R}'_i)$ are occluded in the left view.

We wish to define the cost of matching pixel (x_i, y) in $\mathbf{I}_l$, with pixel (x'_i, y) in $\mathbf{I}_r$, as an aggregation of the lightness differences observed when superimposing the two patches centred at these two pixels. As observed before, not every lightness difference may be taken into account within that aggregation. It is important that, if $(x_i, y) \in \mathbf{R}_i$ and $(x'_i, y) \in \mathbf{R}'_i$, then the difference in lightness between pixels (x_k, y) and (x'_k, y) can be aggregated to the cost of matching (x_i, y) with (x'_i, y), under the following conditions:

- $(x_k, y) \in \mathbf{R}_i$ with respect to the left view and $(x'_k, y) \in \mathbf{R}'_i$ with respect to the right view.
- In order to take into account the lightness differences which are observable when the two patch centres are superimposed, and assuming that the segmented regions are objects standing fronto-parallel to the image plane, we must ensure that $x_k - x'_k = x_i - x'_i$.
- Let $d_i = x_i - x'_i$. The pixel of coordinates (x_k, y) must belong to the patch $\mathbf{P}_i$ centred at (x_i, y), and the pixel of coordinates (x'_k, y) must belong to the patch $\mathbf{P}_i^{(-d_i)}$ centred at (x'_i, y).

Therefore, we can express the cost of matching pixel (x_i, y) with pixel (x'_i, y) as follows:

$$c(x_i, x'_i, y) = \frac{1}{\left|\mathcal{A}(x_i, x'_i, y)\right|} \sum_{(x,y,d)\in\mathcal{A}(x_i,x'_i,y)} \left|\mathbf{I}_l[x, y] - \mathbf{I}_r[x - d, y]\right| \tag{2.2}$$

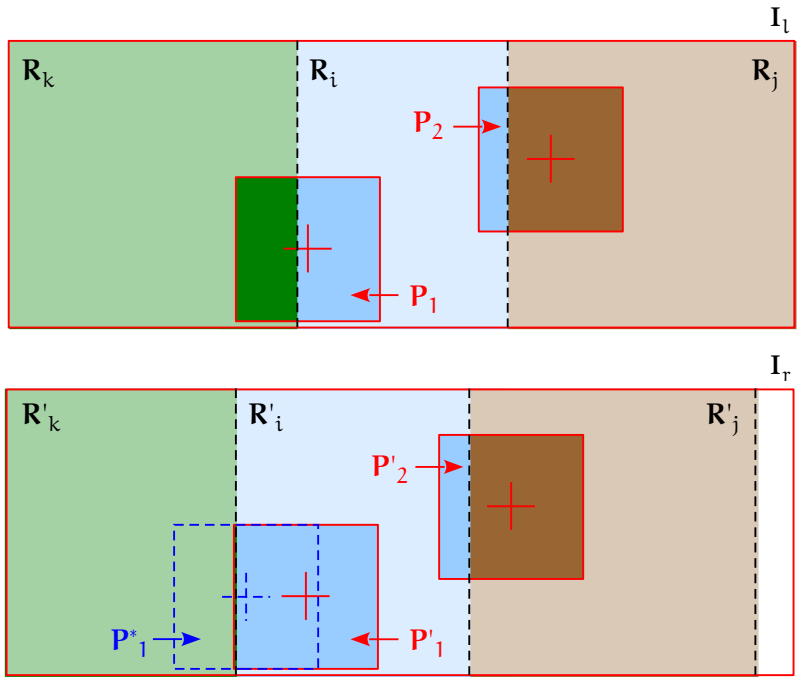


FIGURE 2.3: Scheme illustrating the aggregation consistency problem when patches overlap different objects for the homogeneous case.

where the *aggregation support* is the set of voxels defined as:

$$\mathcal{A}(x_i, x'_i, y) = \{(x, y, d_i) \,|\, (x, y) \in (\mathbf{P}_i \cap \mathbf{R}_i) \wedge (x - d_i, y) \in \mathbf{R}'_i\} \tag{2.3}$$

Finally, since we assumed that the regions were representing fronto-parallel objects, there exists a unique displacement for which two corresponding regions perfectly superimpose, and thus patch $\mathbf{P}_i$ may be discarded from equation 2.3. This type of aggregation being solely controlled by the left and right image segmentations, will prove essential when computing *regional disparities*, a concept which will be developed in chapter 5. In situations where the fronto-parallel assumption would not be valid, the patch could serve as a remedy to the problem posed by tilted regions. More complex methods of tackling this will be presented in the following sections.

Case 2: Homogeneous regions Consider the scenario illustrated in figure 2.3. Here, the regions are homogeneous. The only way to distinguish them is by observing their colour and the only source of information for measuring disparities comes from the frontiers of the regions. In practice, some singularities stemming from the contours are always perceivable near region boundaries. Therefore, by using the aggregation model elaborated for case 1, the aggregation support assigned to the centres of patch $\mathbf{P}_2$ and $\mathbf{P}'_2$ would yield the minimal aggregation cost as desired, because the region frontiers overlap substantially. However the contour singularities

observed in $\mathbf{I}_l$ across $(\mathbf{P}_1 \cap \mathbf{R}_i)$ do not superimpose well within patch $(\mathbf{P}'_1 \cap \mathbf{R}'_i)^{(d_1)}$. This is because the contour perceived in $(\mathbf{P}_1 \cap \mathbf{R}_i)$ does not represent the physical border of $\mathbf{R}_i$, but an occlusion border caused by region $\mathbf{R}_k$. A better aggregation cost would be found by superposing the centres of patches $\mathbf{P}_1$ and $\mathbf{P}^*_1$ for a disparity equal to the one of $\mathbf{R}_k$. Attributing the disparity of $\mathbf{R}_k$ to the centre of $\mathbf{P}_1$ will result in what is commonly referred to as a *fattening effect*.

In fact, the computation of disparities from the contours described in figure 2.3 is open to interpretation, so we shall not seek a method for handling that particular extreme configuration. Nonetheless, this observation is potentially useful, because most images in new datasets comprise both textured and non-textured areas. Using patches within the aggregation scheme presented for this case study suggests that some fattening effect may occur when the aggregation support contains only singularities from a contour which is not the physical frontier of the region under consideration. In order to avoid fattening artefacts, we will propose methods to prioritise the disparity measures obtained within a given region, according to the origin of the singularities.

2.1.3 Inclination and disparity planes

We are now interested in expanding the aggregation domain. In the previous example, we showed that the largest and most relevant aggregation domain, when evaluating a match between two pixels, could be determined in function of the disparity, by the intersection of the regions of origin of these pixels, and under the condition that these regions segment all the objects in the scene and that these objects are positioned fronto-parallel with respect to the image planes of the stereo cameras. Whilst the fronto-parallel assumption has precise applications for low-baseline stereo imagery, it is not appropriate when regions are significantly tilted. For instance, the intersection between two corresponding regions could be large, but only a very small portion of that intersection would be in phase with respect to the image contents. In reality, costs from good and bad superimpositions would be mixed together during the aggregation phase, which would reduce the relevance of the aggregated costs. Therefore, increasing the scale of the aggregation domain requires inspection of the superimposition costs for different disparities.

The assumption that a region is planar is more flexible than the fronto-parallel one and has inspired some approaches using disparity plane fitting. One the one hand, [Sinha et al., 2014] propose resorting to keypoints matched between the images of the stereo pair and compute hypotheses, consisting of disparity planes sweeping some parts of the image. These hypotheses are then used to control the estimation of the full disparity map. On the other hand, [Bleyer and Gelautz, 2005] and [Yang et al., 2009] compute initial disparity measures and cluster the different parts of the images into meaningful regions. They then calculate a plane equation for every region by means of least-squares estimation derived from initial disparity measures, weighted according to their likelihood. All these approaches however require some

initial disparity measures, which are generally obtained using standard patch-based correlation or using keypoint descriptors. A first step towards improving the regional consistency of these disparity plane estimations, would constitute the retention of the disparity plane which yields the optimal superimposition between two corresponding regions, but not the disparity plane which minimises the error with respect to initial disparity measures. To proceed, it would be necessary to superimpose the image contents of any region of $\mathbf{I}_l$ with its corresponding region in $\mathbf{I}_r$, after having performed on the latter, the affine transformation being induced by the estimated disparity plane (see section 1.1.2). However, this would be computationally more expensive and would still necessitate initial disparity measures to make disparity plane proposals. And finally, the model would still be approximate when the objects appearing in the scene are not planar.

In the following subsection, we are going to describe a structure named "disparity space volume" allowing the study of any kind of superimpositions (tilted and fronto-parallel).

2.1.4 Disparity Space Volumes

The *disparity space volume*, abbreviated by DSV, is a stack of stereo image superimpositions, obtained for increasing intensities of left-to-right image displacements. If the superimpositions are characterised by the absolute differences of the image brightness, then, given the image coordinates (x, y) in the left view and the disparity d at which a superposition is tested, the DSV is defined by equation 2.4, as:

$$\mathbf{D}[x, y, d] = |\mathbf{I}_l[x, y] - \mathbf{I}_r[x - d, y]| \tag{2.4}$$

The DSV should facilitate the analysis of the stereo superimpositions in 3D, and this would be ideal when processing a region for which the depth function cannot be approximated by a simple model. But first and foremost, it is essential to know how the patterns encountered in a DSV may be interpreted.

Analysis of disparity space volumes

Now, let us look at figure 2.4. Figures 2.4(***c***) and 2.4(***d***) show a pair of stereo images originating from Middlebury 2014 database. In red, two corresponding scanlines at $y = 60$ are highlighted. The *disparity space image* or DSI displayed in figure 2.4(***b***) is a slice of the DSV, representing the way superimposition costs evolve along the scanline $y = 60$ for all tested disparities. For zones **B** and **D**, which contain texture, human vision is able, without much difficulty, to localise the minima which correspond to relevant superimpositions. It is much harder to discern these minima for zones **C**, **E** and **G** without resorting to a preliminary shape-based aggregation that takes account of neighbour scanlines, as shown in figure 2.4(***a***). Finally, zone **A** is the most

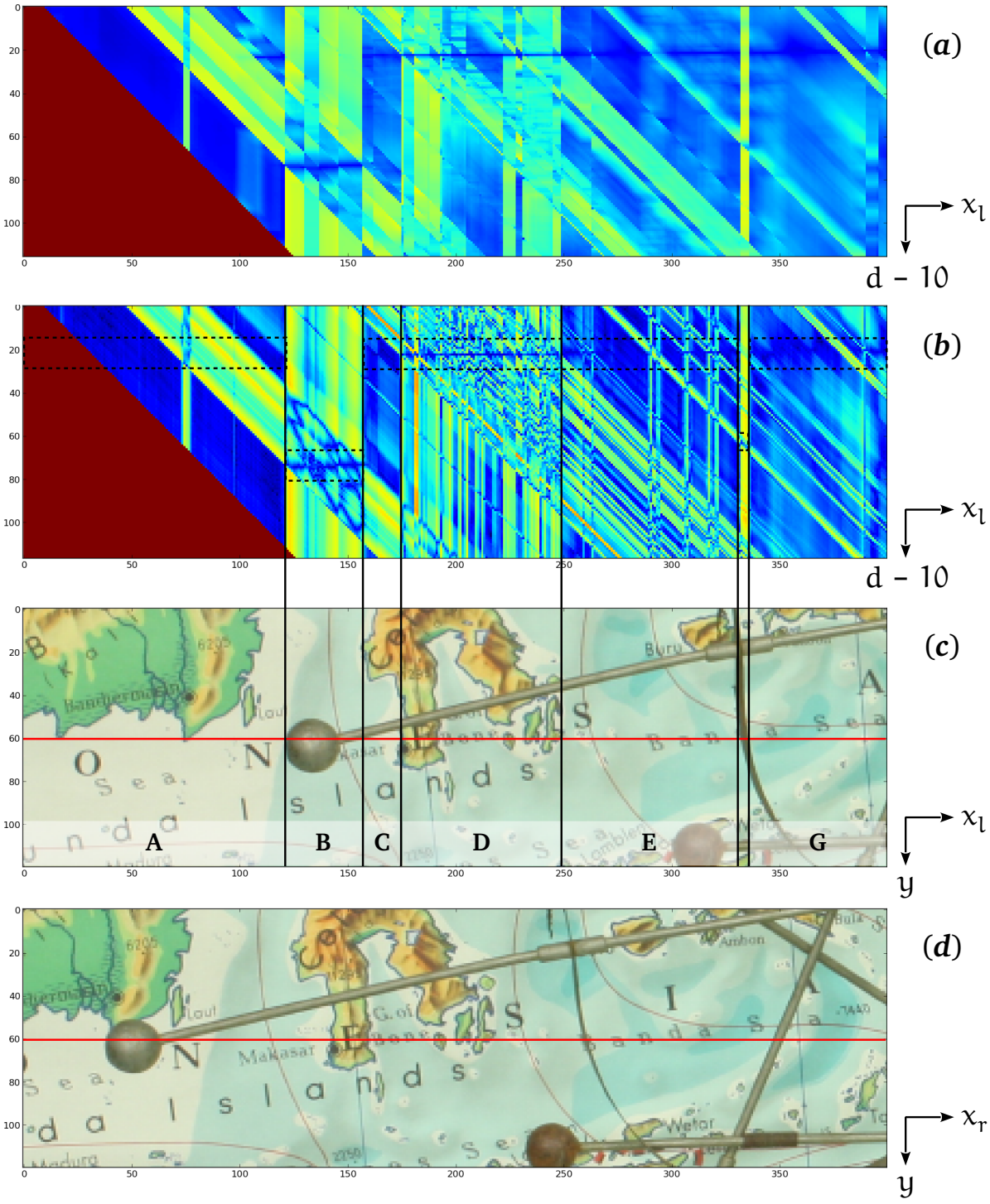


FIGURE 2.4: Disparity Space Images generated when warping the scanlines of `AustraliaP` stereo images (c) and (d), shown in red. The raw DSI is visualised in (b) using a jet colormap, i.e. with the lowest costs represented in blue, and the highest costs represented in red. The vertical lines in black help identify the regions traversed by the scanline in the left view, while the dotted rectangles highlight the areas of the DSI where the actual superposition should take place. The costs of the DSI shown in (a) stem from a shape-adaptive aggregation using patches of 11×11 pixels (see section 2.1.2, equation 2.2) with fixed displacement. One can readily appreciate how this preliminary aggregation solves some of the ambiguities.

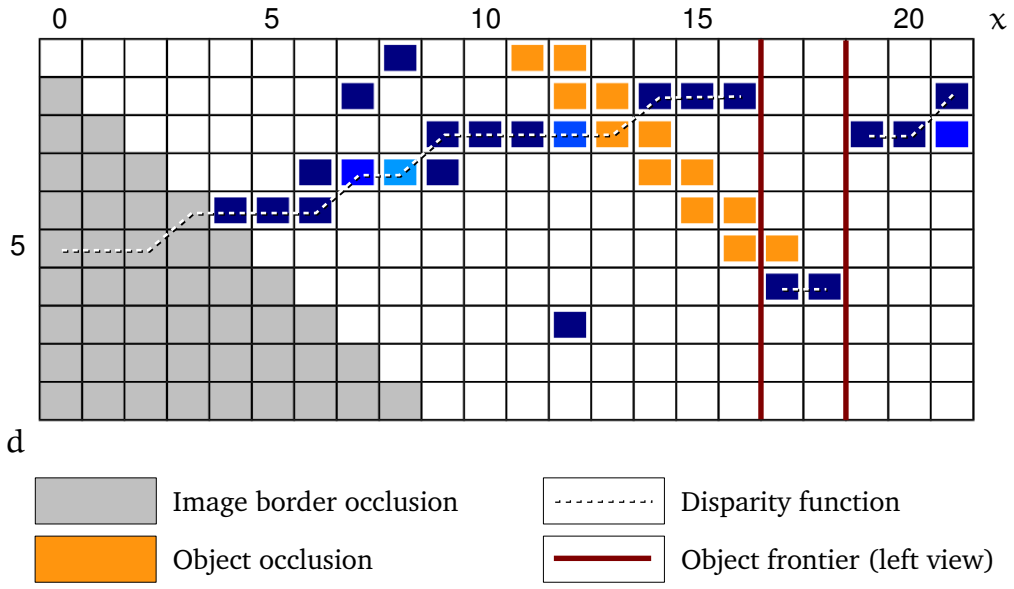


FIGURE 2.5: Illustration summarising the main structures to anticipate when analysing disparity space images. Non-visible registrations will occur across occlusion traces and discontinuities of disparities may occur upon crossing the physical frontier between two objects in the scene.

difficult to interpret for the three reasons which makes depth map computation a complex problem:

- the scanline in zone **A** traverses a relatively homogeneous region. Therefore, the shape-aggregation constrained by our patch of size 11×11 pixels does not suffice to solve the ambiguities. From $x_l = 30$ to $x_l = 60$, it is impossible to tell where the superimposition occurs.
- with respect to the right view, a fairly significant area of zone **A** is occluded by the sphere appearing in zone **B**. We can see that the occlusion trace have the same width as zone **B**, in zone **A** of the disparity space image. This trace is characterised by high superposition costs. Furthermore, if a point with abscissa x_i in the left view is assigned disparity d, then the pixels contained in the set $\{(x_i - \tau, d - \tau) \mid \tau > 0\}$ belong to the occlusion trace of the DSI attributed to the scanline of that point.
- the left-hand side of the left view undergoes an image border occlusion. This is the part of the disparity space image shown in red. Again, no disparities can be measured in this area.

Figure 2.5 illustrates and summarises the main patterns which can be observed from disparity space images. Early methods of stereo analysis exploited the structures of the filtered disparity space images, for each scanline of the image. These methods relied on warping techniques, which are classified as one of the semi-global approaches for depth map computation.

Scanline warping

Scanline warping provides a way of finding a connected path through the DSI which travels past the superimposition shadow. It is based on a variant of dynamic time warping [Müller, 2007], which will be described in this paragraph. For the sake of simplicity, the DSI is first converted to a Warping Space Image, or WSI, which is a slice of a Warping Space Volume as described in table 2.1. An example of such image is given in figure 2.7(***a***).

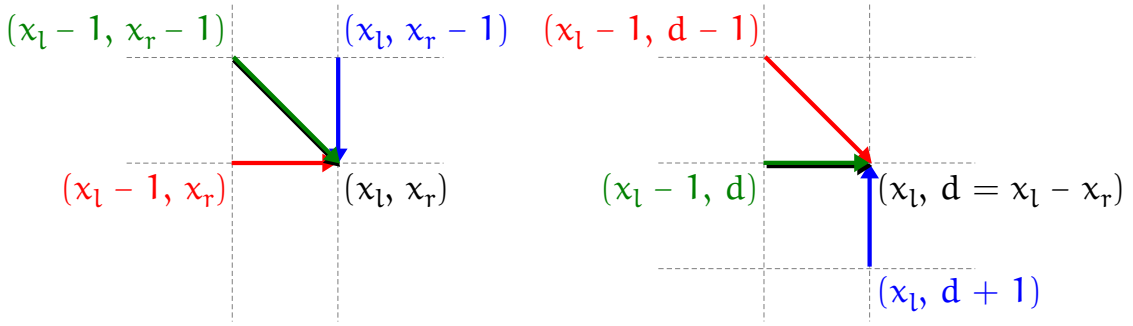
Problem definition In the case of stereo, a valid warping path $\Gamma_y : \{1, \dots, n\} \to \{(x_l^{(i)}, x_r^{(i)})\}_{i=1}^n$ has the following characteristics:

- The initial point of the path $(x_l^{(1)}, x_r^{(1)})$ must satisfy $x_r^{(1)} = 0$, so that the first pixel of the right image scanline must be matched with one pixel of the left image scanline.
- Likewise, the final point of the path $(x_l^{(n)}, x_r^{(n)})$ must satisfy $x_l^{(n)} = W - 1$, where W is the width of the image, in order to enforce a match of the rightmost pixel of the left image scanline, to one pixel of the right image scanline.
- The path must evolve in such a way that only three transitions can occur: either both abscissa are incremented or only one abscissa is incremented. Doing so guarantees that all $x_l \geqslant x_l^{(1)}$ have been visited, likewise all $x_r \leqslant x_r^{(n)}$, and that the *ordering constraint* applies, i.e. $x_l^{(i)} \leqslant x_l^{(i+1)}$ and $x_r^{(i)} \leqslant x_r^{(i+1)}$ for all $i \in \{1, \dots, n-1\}$. These transitions are illustrated in figure 2.6.
- The cost of a path is evaluated as follows:

$$c(\Gamma_y) = \sum_{i=0}^{n} \underbrace{W\left[x_l^{(i)}, x_r^{(i)}, y\right]}_{\text{superimposition cost}} + \sum_{i=1}^{n} \underbrace{\Upsilon(d_{i-1}, d_i)}_{\text{occlusion penalty}} \tag{2.5}$$

  such that the occlusion penalty $\Upsilon(d_{i-1}, d_i)$ between two consecutive disparities $d_i = x_l^{(i)} - x_r^{(i)}$ and d_{i+1} adds a tiny contribution ξ to the cost as soon as the disparities differ, i.e. when two consecutive pixels along a scanline merge into one pixel in the other scanline. Such a phenomenon is often attributed to an occlusion, although it may be due to the quantisation of disparities across tilted regions. The real purpose of the penalty in this particular case, is to hinder non-relevant variations of disparities across homogeneous regions.

We seek a path $\Gamma_y^\star$ for which the cost $c(\Gamma_y^\star)$ is the least amongst all valid paths. The solution consists of computing a *distance function* from the border of the Warping Space Image, which contains the set of valid initial points. This distance function depends on two factors: the relief traversed, which is composed of the superimposition costs, and the available path directions from any point of the relief. The ending point of the path $\Gamma_y^\star$ lies in the target WSI border, consisting of the set of valid ending points, and has the minimum cost or distance.



Aggregation directions in the **Warping Space Image**

Aggregation directions in the **Disparity Space Image**

FIGURE 2.6: Directions driving the aggregation of costs within the accumulator array for both the warping and the disparity space images. The colours highlight the correspondences between these directions across the two proposed models. Transitions shown in red symbolise *pixel occlusions* occurring in the right view. Transitions shown in blue symbolise *pixel occlusions* occurring in the left view.

A solution based on dynamic programming Let $\mathbf{A}$ be called the *accumulator image*, having the same dimensions as the WSI generated for the scanline of ordinate y, and containing the values of the distance function as shown in figure 2.7(***b***). All distances are initially set to $+\infty$. Given the constraint on the path evolution, the distance from the set of initial points to a particular coordinate (x_l, x_r) of $\mathbf{A}$ can be expressed by recurrence as:

$$\mathbf{A}[x_l, x_r] = \boldsymbol{W}[x_l, x_r, y] + \min \begin{cases} \mathbf{A}[x_l - 1, x_r - 1] & \text{(Match, } \Delta d = 0\text{)} \\ \mathbf{A}[x_l - 1, x_r] + \xi & \text{(Left scanline pixels merge, } \Delta d = +1\text{)} \\ \mathbf{A}[x_l, x_r - 1] + \xi & \text{(Right scanline pixels merge, } \Delta d = -1\text{)} \end{cases} \tag{2.6}$$

paying particular attention to the initialisation, when $x_l < 1$ or $x_r < 1$:

$$\mathbf{A}[x_l, x_r] = \begin{cases} \boldsymbol{W}[x_l, x_r, y] & \text{if } x_r = 0 \\ +\infty & \text{otherwise} \end{cases}$$

The argument of the minimisation in equation 2.6 encodes, for a given point of coordinates (x_l, x_r), the previous coordinate of the path of minimum distance which leads to the point of coordinates (x_l, x_r). This information is essential when performing *backtracing*, i.e. the process of finding the complete optimal path which leads to the chosen point of the WSI. Finally, note by comparing equations 2.6 and 2.5 that if $(x_l^\star, x_r^\star)$ is the ending point of $\Gamma_y^\star$, then $\mathbf{A}[x_l^\star, x_r^\star] = c(\Gamma_y^\star)$. As a result, the optimal warping path is recovered using backtracing from the match minimising $\mathbf{A}[x_l, x_r]$ for $x_l = W - 1$.

Known issues Although the shadow detection presented in figure 2.7(***c***) is useful, scanline warping is a far from perfect method. Firstly, the ordering constraint applied to the whole scanline, is an assumption which retains minimal validity with respect to current datasets. In the example of figure 2.4, the object lying between zones **E** and **G** violates the ordering constraint along the scanline. This explains the fact that its disparity could not be captured. In fact, the ordering constraint should not be neglected, but should be applied on a regional basis, where its scale remains consistent. Secondly, the search for the optimal warp using the end matching points appears to be a source of potential instabilities: for example, what would happen if the right-hand side of the left view is occluded in the right view? Thirdly, the cost associated with the ideal warp between two corresponding scanlines, integrates the superimposition costs occurring across occluded areas. We observed that such costs take fairly high values. Therefore, if the ideal warp travels past a large number of occluded points, its overall accumulated cost is likely to be more significant than that of an optimal warp found by minimising equation 2.5. As a result, the optimal warp would then be different from the ideal warp. Zone **A** in figure 2.7(***c***) illustrates this phenomenon. Finally, we should be aware that the warping path, when x_r remains fixed, does not correspond to the actual disparity function, but merely encodes the fact that x_l is occluded in the right view. Figure 2.8 illustrates that observation.

Using prior information In their work, [Bobick and Intille, 1999] attribute occlusions to the portions of the path, for which only one of the two scanline abscissa has been incremented. The value added to the current path distance when such an occlusion occurs, does not take into account the superimposition cost, but the occlusion cost ξ only. In order to be coherent, an occlusion cost which is higher than the typical cost associated with a valid match, should be chosen. Nonetheless, the authors acknowledge that without any intervention, the optimal warp remains very sensitive to ξ, in particular when occlusions are significantly large. They show that, forcing the path to visit reliable matches limits the impact of the occlusion cost. Furthermore, they adapt the occlusion cost dependent upon whether an edge in the left or right image is traversed, as a method of improving tolerance of the realisation of an occlusion near edges, which constitutes a sensible assumption. Nevertheless, the need for ground control points demonstrates that these global algorithms require good initialisations to function properly.

In short Scanline warping is a method of aggregating superimposition costs which arise from different disparities. In its standard formulation, only one aggregation cost yields the full disparity function, but it would be more relevant to perform the operation at a regional level. The occlusion phenomenon requires special care. Finally, as the preliminary shape-constrained aggregation has demonstrated, the need for an aggregation that works across different scanlines is real. Ideally, it should be possible to scale the process described here for one scanline, to work across multiple scanlines, but the algorithm extensions to 3D are absolutely not straightforward and

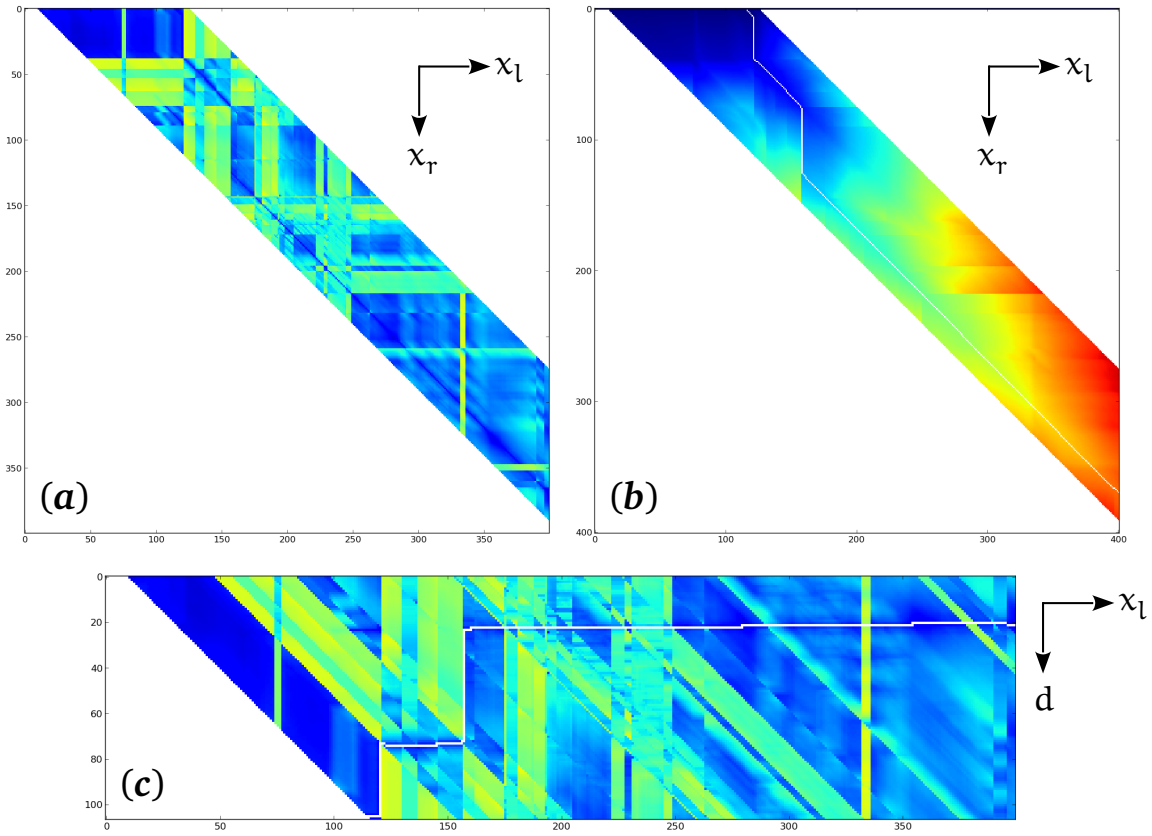


FIGURE 2.7: Warping the scanlines highlighted in figure 2.4. (a) Warping Space Image representation of the DSI in figure 2.4(a). The part of the array shown in white denotes points allocated an infinity cost value. (b) Accumulator array resulting from the aggregation procedure retained for the warping task. The line in white shows the path that traverses the array with minimum cost. (c) Warping path projection onto the original DSI.

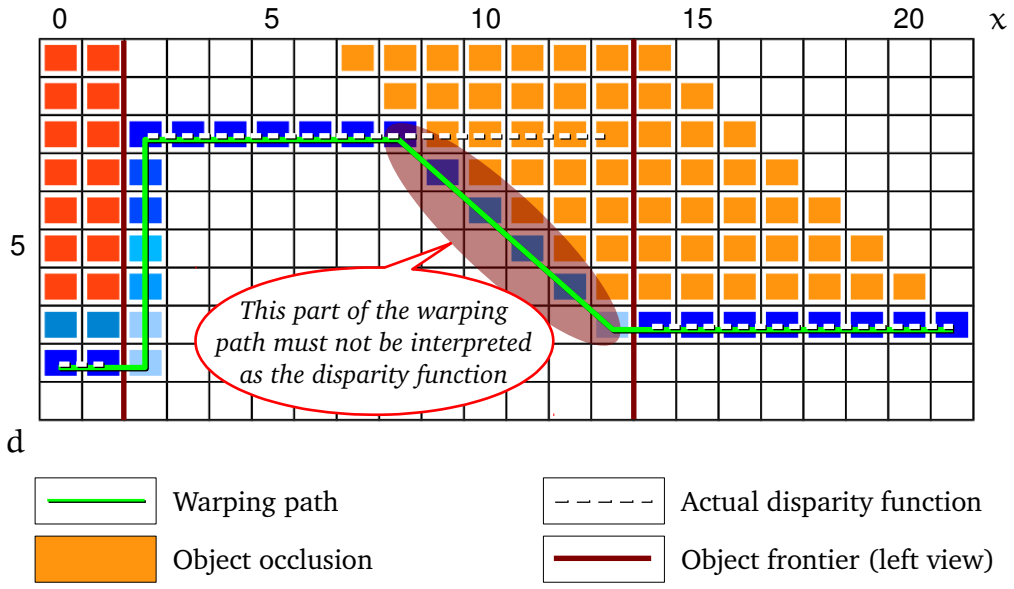


FIGURE 2.8: Comparison between the warping path and the actual disparity function being searched for. When $x_r = x_l - d$ remains on a portion of the warping path, that portion of the path is no longer in phase with the disparity function, but merely encodes that x_l is occluded in the right view.

still constitute an active field of research.

Inter-scanline warping

One of the early methods of exploiting the 3D structure of a disparity space volume was proposed by [Ohta and Kanade, 1985]. For each pair of corresponding scanlines, their algorithm consisted of warping the two associated sequences of image contour points, so that the resulting warpings would ensure disparities remained coherent along the image contours crossing different scanlines. To proceed, the problem was separated into two stages: first, the search for intra-scanline warping paths using a technique similar to that discussed at the beginning of this section; second, an aggregation of the accumulator arrays computed for each scanline. This aggregation provides the inter-scanline consistency constraint required for the selection of the optimal warping paths for each scanline.

A recent algorithm which exploits warping and inter-scanline consistency is the semi-global matching or SGM algorithm, proposed by [Hirschmüller, 2008]. Instead of searching for a unique warp between the horizontal scanlines, the algorithm seeks, for each pixel of the image plane, a series of warping paths originating from the image border and ending at the pixel of interest. The directions of the image lines along which the warping procedure takes place are uniformly sampled. The superimposition costs along all these optimal warps are aggregated to the pixel of interest for every valid disparity, and are stored inside an accumulator volume. The arguments minimising the aggregation cost along the disparity axis of the accumulator volume

yields the disparity map. Furthermore, the author shows that the SGM algorithm approximates the solution of an energy minimisation problem which constitutes the roots of most of the global approaches in stereo. These will be addressed in section 2.3. Finally, it should be noted that SGM often forms an essential ingredient in modern approaches, such as [Sinha et al., 2014] or [Zbontar and LeCun, 2015].

Inter-scanline warping methods therefore exploit combinations of 2D warpings. They also require a sampling strategy to determine the sequence of pixels in the left image which has to be warped, i.e. pixels along a contour, or along a particular trajectory.

2.1.5 Preliminary conclusions

The cost of a warping path is nothing other than an aggregation of the superimposition costs found in a disparity space volume. Therefore, in the case of standard scanline warping, the cost associated with the superposition of (x_l, y) in $\mathbf{I}_l$ with $(x_l - d, y)$ in $\mathbf{I}_r$ equals the cost of the optimal warping path going through (x_l, d) in the DSI associated with scanline y. The mechanism proposed by the SGM is more sophisticated, in the sense that the cost associated with the same superposition is related to the costs of the warping paths ending at (x_l, y, d) in the DSV. The fact that the warping algorithms operate at a global image scale, means that the handling of disparity discontinuities demands particular attention. In the case of the standard warping algorithm, the discontinuities can be modelled only by a sequence of occlusions. In the SGM, the handling of discontinuities is embedded in the computation of the accumulator arrays.

Now, in the light of the shape-based aggregation studied at the beginning of this section, we notice that the problem with the (semi-)global strategies is that the aggregation of costs has no meaning with respect to the regions composing the image of interest: indeed the aggregated cost of a region lying at one end of the image may easily depend on the superimposition cost computed for a region localised at the other end of the image. If we know which regions constitute the objects in the scene, we can apply a warping on a regional basis, thereby simplifying the algorithms so that no handling of discontinuities is needed. Furthermore, we could also constrain the warpings so that they are limited to the superimposition of regions which truly match.

The measurement of disparities will be a very important aspect of this dissertation and will be thoroughly discussed in chapter 5. The observations made in this section will be reviewed when necessary. Another important aspect of the measurement process will be to determine whether it is possible to measure the disparity of a given pixel. In the following sections, we provide simple yet efficient techniques to detect and process occlusions. For any pixel left without a disparity measure, it will be necessary to estimate its disparity value, which we would then be able to determine in function of the disparities measured in the region to which such a pixel belongs.

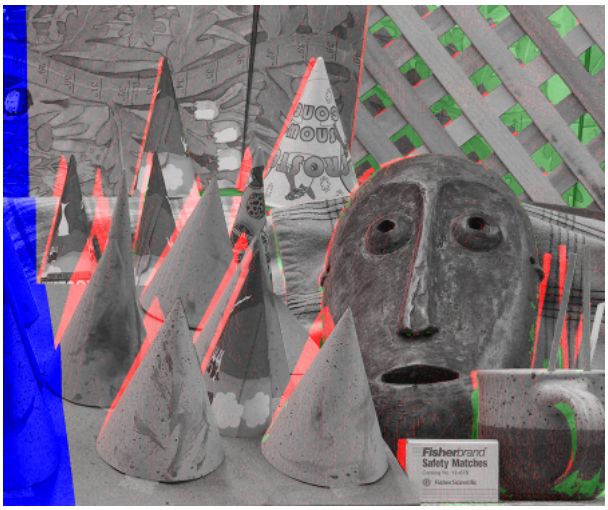


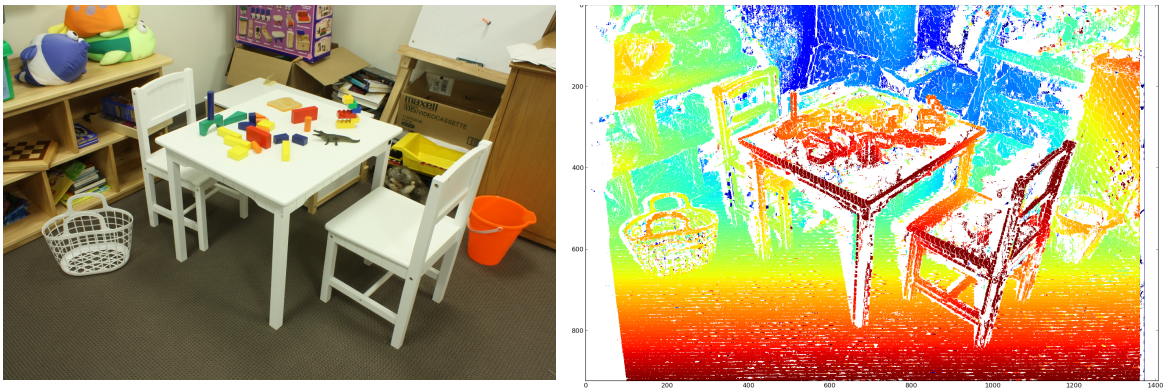
FIGURE 2.9: Left view of the `Cones` scene (see figure 2.1) highlighting pixels which are occluded in the right view. The computation of the occluded areas originates here from the ground truth provided with the Middlebury 2002 benchmark. The areas shown in red correspond to *object occlusions* and are hidden in the right view by a nearby object located closer to the camera. The areas shown in blue correspond to *image border occlusions* occurring because the corresponding pixels in the right view lie outside the definition domain of the image plane. The areas shown in green comprise the pixels for which the ground truth provides no measure of depth.

2.2 Detecting and handling occlusions

When introducing this chapter, we pointed out that it is not possible to find a match for pixels appearing just in one view of the stereo pair. In fact, pixels undergoing an occlusion in one of the views are usually allocated a high superimposition cost for their actual disparity. This is quite inconvenient when deriving disparities from image superimpositions and furthermore necessitates a careful reasoning to estimate the disparities. Additionally, to find the exact image areas which are occluded, as in figure 2.9, the availability of the actual disparity map is required, yet that is precisely what is being sought. As a result, the method of handling the occlusions is often dependent on the chosen approach for estimating the disparity maps. Nevertheless, some dedicated techniques are available to cope with the occlusion phenomenon. It is to these techniques that we devote this section.

2.2.1 Local handling

We shall concentrate here on two approaches for handling occlusions at the pixel level: first, cross-checking, which aims to establish matches which are consistent with respect to the disparity maps measured for the left and right views of the stereo pair; second, adaptive neighbourhood methods, which aim to estimate disparities of pixels occluded in one view, given the disparities of non-occluded neighbour pixels.



(*a*) (*b*)

FIGURE 2.10: Pixels satisfying the cross-checking criterion for (*a*) the left view of `PlaytableP` are represented with their disparities in (*b*). Other pixels appear in white and mainly cover both border- and object-occluded areas as well as homogeneous areas across the table and the chair. The DSV from which the disparity map arises has been obtained using a shape-constrained aggregation.

Cross-checking

Consistent matches between the left and right views can be enforced by an efficient technique called *cross-checking* [Fua, 1993], which ensures that every pixel has at most one correspondence in the other image of the stereo pair provided that correspondences are *undirected*. This condition is unlikely to be the case for pixels appearing only in one of the views: they may find a match which minimises the superimposition cost over all the possibilities in the other view, but that particular match may have a better, non-occluded correspondence in the initial view. More formally, a correspondence between (x_l, y) in $\mathbf{I}_l$ and (x_r, y) in $\mathbf{I}_r$ satisfies the cross-checking criterion if and only if :

$$x_l = \arg\min_x \boldsymbol{W}[x, x_r, y] \ \wedge \ x_r = \arg\min_x \boldsymbol{W}[x_l, x, y] \tag{2.7}$$

Correspondences which do not satisfy the cross-checking criterion are said to be invalid. Occluded points belong to this group, as do points lying in homogeneous areas, as shown in figure 2.10. Cross-checking can therefore be used as a powerful binary indicator of disparity measurement reliability. Further applications of cross-checking will be addressed in our methods.

However cross-checking does not systematically prune the fattening artefacts which occasionally appear across the homogeneous image areas being occluded in one of the stereo views. Consider the illustration presented in figure 2.11. In the two stereo images, $\mathbf{R}_i$ corresponds to $\mathbf{R}'_i$ and $\mathbf{R}_k$ corresponds to $\mathbf{R}'_k$. All these regions are fully homogeneous, with the exception of the little halo surrounding the contour separating the two regions in each view. The hatched area in the left view represents the parts of $\mathbf{R}_i$ occluded in the right view. Pixel $\mathbf{p}_1$ is therefore occluded in the right view. Matching $\mathbf{p}_1$ with $\mathbf{p}'_1$ would yield the actual disparity of pixel $\mathbf{p}_1$,

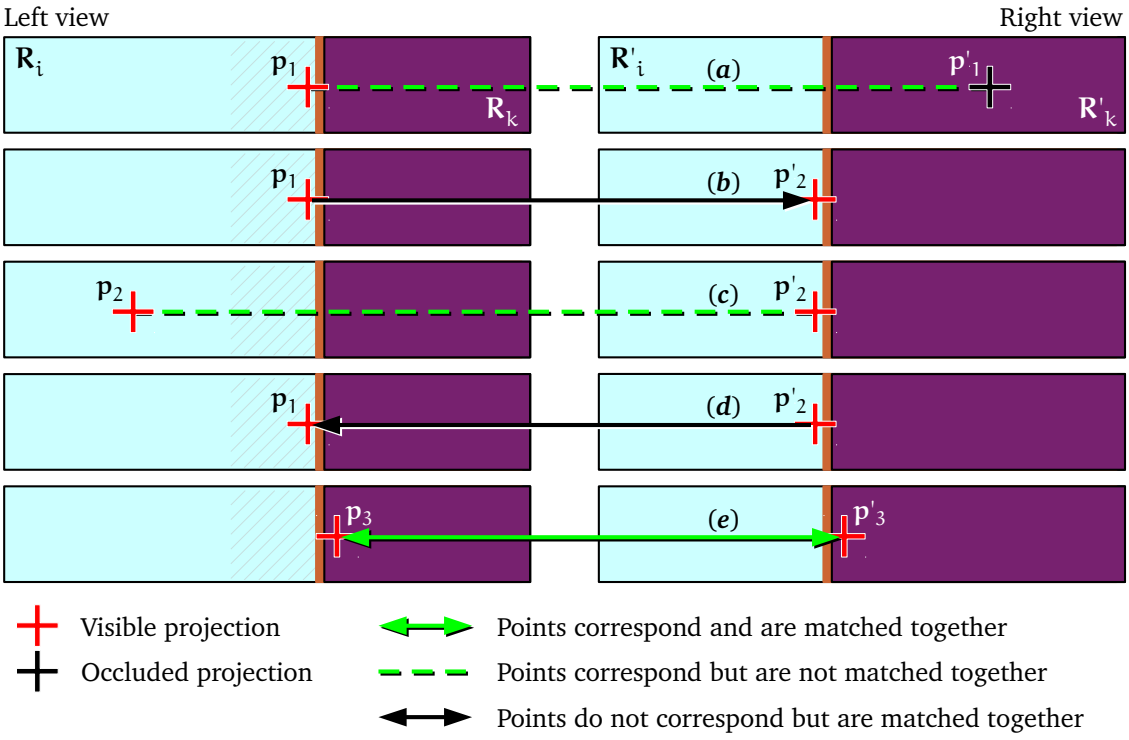


FIGURE 2.11: Cross-checking and fattening artefacts.

but the superimposition cost related to correspondence (***a***) is meaningless. Because of the halo surrounding the contours, the pixel which constitutes the best match for $\mathbf{p}_1$ in the right view is $\mathbf{p}'_2$. $\mathbf{p}'_2$ is not occluded in the left view, and genuinely corresponds to $\mathbf{p}_2$. But due to the halo effect observed in the vicinity of $\mathbf{p}'_2$, the match minimising the superimposition cost in the left view is $\mathbf{p}_1$. Due to correspondences (***b***) and (***d***), the cross-checking criterion is satisfied when pairing pixels $\mathbf{p}_1$ and $\mathbf{p}'_2$. The measured disparity is identical to that measured for correspondence (***e***), which explains the fattening effect observed within the disparity map, even after the bad measures have been discarded by the cross-checking criterion.

Adaptive neighbourhood

The occlusion state of a single pixel is either visible or occluded. Intuitively, the disparity of an occluded pixel should be estimated by looking at its neighbour pixels, within the region to which it belongs. Indeed, sufficiently large regions are prone to undergo semi- rather than total-occlusions and the pixels remaining visible in both images of the stereo pair should share consistent disparities. Without resorting to segmentation, shiftable windows [Kang et al., 2001] have been used to attenuate the occlusion impact on the superimposition costs by testing different patches *not* centred at the point of interest. The purpose of this operation was to increase the chances of having an aggregation support able to capture the superimposition of the region to which the occluded pixel belongs.

The work of [Huq et al., 2013] builds on the cross-checking and investigates a filling strategy for occluded areas, called *Neighbour's Disparity Assignment*. Should the disparity map be estimated for the left view of the stereo pair, it can be observed from figure 2.9 that pixels belonging to *object* occlusions must retrieve their disparities from the vicinity of their left, whereas pixels belonging to *border* occlusions must refer to the vicinity of their right. In its standard formulation, NDA searches for the nearest visible neighbour along the scanline and allocates its disparity to the occluded pixel of interest. It is important to notice that this model assumes the patches being completed with a disparity are positioned fronto-parallel to the camera. The same authors have proposed an extension of that model which is based on a regional handling of the occlusion phenomenon.

2.2.2 Regional handling

Given the region to which a pixel belongs, estimating its disparity amounts to solving an interpolation problem. The complexity of that problem depends, of course, on the character of the scene. For example, if all regions stand fronto-parallel to the image plane, all pixels lying in the same region, including those occluded, must share the same disparity, and the interpolation is therefore straightforward. State-of-the-art methods tend to favour plane equations, like the segmentation-based approach in [Huq et al., 2013] which resorts to least-squares fed by control points, lying in the vicinity of the occluded pixel and belonging to the same region. Later in this text, we will consider other interpolation alternatives which offer more flexibility and which are based on linear estimation.

Border occlusions are slightly more problematic, in that total occlusions are more likely to occur. In fact, when an entire object disappears from one of the views, there is no available clue as to how to recover its depth. However, it can be helpful in other cases to fuse the totally occluded regions with the visible regions lying in their vicinity to obtain some clues for interpolation. This fusion, of course, is not straightforward and we shall propose an algorithm dedicated to that task in the second part of this dissertation.

Many stereo algorithms still handle occlusions *on the fly* using global estimation procedures, which we will describe in the following section.

2.3 Estimating disparities

Based on empirical observations, a disparity map of an outdoor or indoor scene is nothing other than a piece-wise continuous topographical surface. The discontinuities of that disparity function should only occur close to edges delineating the boundaries of objects in the scene, whilst the

disparities conveyed by the function should yield a warp between the two stereo images which minimises the superimposition cost as much as possible.

2.3.1 Energy-based formulations and limits

That observation has led to a class of stereo algorithms driven by energy minimisation, for which the objective function to minimise is expressed by equation 2.8.

$$\Theta(\mathfrak{D}) = \underbrace{\sum_{(x,y)} U\left(\mathbf{I}_l\left[x,y\right] - \mathbf{I}_r\left[x-d,y\right]\right)}_{\text{Superimposition consistency term}} + \underbrace{\sum_{(x,y),\,(x',y')\in\mathcal{N}_{(x,y)}} P(d-d')}_{\text{Smoothness term}} \tag{2.8}$$

d and d' represent the disparities $\mathfrak{D}\left[x,y\right]$ and $\mathfrak{D}\left[x',y'\right]$ respectively, while $\mathcal{N}_{(x,y)}$ denotes the set of pixels lying in the direct neighbourhood of (x,y). The function U penalises a disparity which yields a bad pixel superposition while the term P penalises neighbour disparities inducing a discontinuity on the disparity map. In general, P has a dependency on the view for which the disparity map is being estimated, since discontinuity penalties must be attenuated near edges. As such, this energy-based formulation of the stereo problem involves finding a *trade-off* between warping consistency and surface regularisation.

The minimisation of $\Theta(\mathfrak{D})$ is typically achieved by one of the following methods. The first category is based on gradient-descent algorithms, for which the penalty term P is driven by the gradient magnitude [Fua, 1993] or the regional boundaries [Aydin and Akgul, 2010]. The second category proceeds from the observation that minimising the energy term in equation 2.8 is analogous to inferring a maximum a posteriori from a Markov Field modelling the relationship between the sought disparities and both the superposition consistency and continuity constraints. The exact solution to this inference problem is described in [Prince, 2012]. It resorts to a graph-cut algorithm seeking a segmentation of the disparity space volume into the two groups of voxels: those lying in front of and those lying behind the surface which encodes the estimated disparity map. The space and time complexity of this algorithm however renders it unattractive for very large numbers of voxels, which is the situation with disparity space volumes generated for HD stereo imagery. Indeed, these volumes can easily reach hundreds of millions of voxels. For that reason, approximation algorithms, such as alpha-expansion [Boykov et al., 2001] and loopy belief-propagation [Sun et al., 2005, Zitnick and Kang, 2007, Yang et al., 2009] are generally favoured to perform that inference. The SGM also belongs to this class of approximation algorithms, and an extension has recently been suggested by [Facciolo et al., 2015].

The first concern is the way occlusions are handled. Again, the smoothness term plays a non-negligible role in the context of preventing abrupt disparity changes, which can result from mismatches across occluded areas. But it is clear that if the surface being occluded is significant

or larger than that visible for a given region, then the superimposition consistency term U is likely influence the final estimate. When an estimate of the occlusions is provided, it is useful to take that into account in order to eliminate the contribution of the term U at each pixel of the left image being occluded. Such an approach has been proposed in [Yang et al., 2009]. It consists of performing several hierarchical refinements to provide U with the estimated location of occluded areas.

Another goal of estimation in depth map computation is the production of plausible disparities across homogeneous regions. For those with constant intensity, the smoothness term of equation 2.8 is key to providing the estimation of the disparity function, since the superimposition consistency term will then only yield null penalties when the matched pixels belong to the same object. Yet, it is necessary to have at least one discriminant feature enclosed within that region, otherwise the prediction of the energy minimisation behaviour will be a problem, if all disparity information arises from the region borders, in particular if the borders represent occlusion contours with respect to one area of the scene.

2.3.2 Dealing with homogeneity

Homogeneous regions continue to be handled imprecisely in current stereo analysis; an inexactitude which constitutes one of the most persistent constraints on dealing with imagery devoid of texture. We shall end this section with some observations that could be taken into consideration in that respect.

Consider the illustrations shown in figure 2.12. In the absence of texture, only the object contours convey the disparity. In theory, any disparity function employing the disparities measured at the contours would be plausible. The question is whether this disparity function should be fully or piece-wise continuous. The answer will depend on the way the segmented regions are interpreted. On the one hand, if regions constitute one single object, then the disparities inside each region should result from an interpolation of their contour disparities. On the other hand, if regions denote disconnected objects in the scene, then it is important to determine for each region whether the contour represents a physical frontier, or whether it is due to an occlusion. Consequently, for the process of disparity interpolation, a region is limited to using the disparities provided by contours which represent its physical frontiers. To date, few methods performing this distinction between physical and occluding contours exist. We can however refer the reader to the work of [Yamaguchi et al., 2012] which, to this end, analyses the structures of junction boundaries.

Before concluding this section, it should be noted that matching region contours to obtain their disparities can, in some cases, introduce a bias to the measured disparity. This bias can be

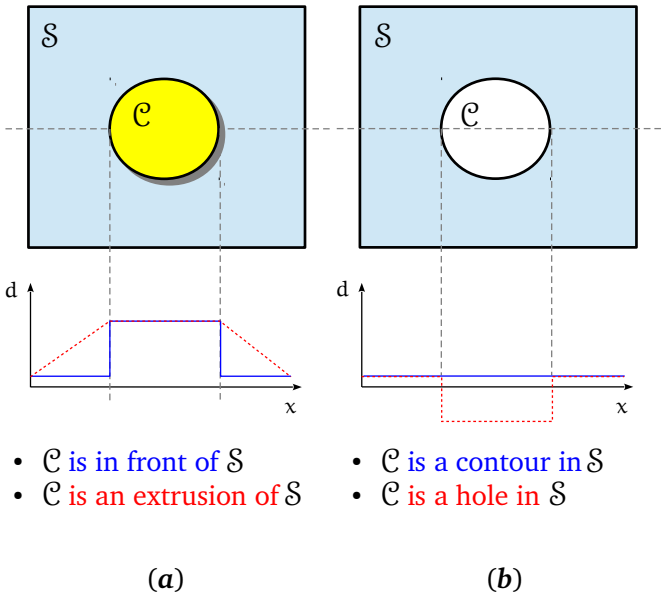


FIGURE 2.12: Estimation of depth from contours. (*a*) The contour separating $\mathcal{C}$ from $\mathcal{S}$ has a higher disparity than does the border of $\mathcal{S}$. We give two interpretations to this observation. Either object $\mathcal{C}$ is spatially connected to object $\mathcal{S}$ and the disparity function has to evolve smoothly, or $\mathcal{C}$ lies in front of $\mathcal{S}$, in which case $\mathcal{C}$ inherits the disparity from the contours between $\mathcal{C}$ and $\mathcal{S}$, whilst $\mathcal{S}$ inherits the smallest of the contour disparities. (*b*) All contours are measured with the same disparity. We can allocate that disparity to both regions $\mathcal{C}$ and $\mathcal{S}$ which would imply that both regions form the same object. Notice that $\mathcal{C}$ might be a region seen through a hole of $\mathcal{S}$. We can deduce no information about that disparity can be said from the contours. We can search for another region having identical colour properties in the vicinity of $\mathcal{S}$ and transfer its disparity to $\mathcal{C}$.

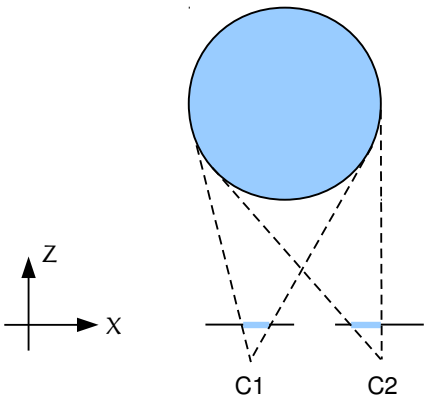


FIGURE 2.13: The self-occlusion phenomenon

explained by the self-occlusion phenomenon, which occurs on rounded objects, as illustrated in figure 2.13. For that reason, we may conclude that the use of disparities arising from region contours requires some reflection in order to determine the appropriate region for employing these disparities within an interpolation process. Finally, due to the self-occlusion ambiguity, disparities measured near the contours of homogeneous regions should be used only, when no disparity clue is available inside the region of interest.

Summary

In this chapter, we introduced depth map computation as a problem of measuring and estimating disparities between two images of a stereo pair. By superimposing the right view onto that of the left, we observed that a superimposition shadow sweeps the image plane as the right view is shifted towards the right. The time at which a pixel in the left view is covered by this superimposition shadow corresponds to its disparity. In order to highlight this shadow, it is necessary to aggregate the individual pixel superposition costs in a meaningful way: the aggregation support of a given pixel should contain only the pixels which are enclosed in the same region and which are not occluded in the other view. This observation led to the definition of an unconstrained regional aggregation support, which is, under the fronto-parallel assumption, identical for all pixels belonging to the same region. We wished to generalise that regional support for other configurations, and for that reason, we introduced the disparity space volume from which we proposed to extract the enclosed superimposition surfaces, both at a regional level and across multiple disparities. To achieve this, distance functions, similar to those used for the classic warping, could be employed.

It should be borne in mind that depth estimation is an inverse problem, and as such, requires an estimation model, to predict the disparity values where pixels are occluded and where regions are homogeneous. In the case of homogeneous regions, we demonstrated that it is essential to determine to which object or region a contour belongs, before using its disparity. Furthermore, we devised a prioritisation for the disparity clues, which imposes that disparities along contours and near homogeneous areas should be considered, only when no internal clue is available.

In conclusion to this second chapter, it should be remembered that regions bring pertinent information to the measuring and estimation steps required for computing depth maps. The regions to which we referred were delimiting object boundaries. However, finding a method of producing a segmentation which truly segments the objects in a scene still constitutes an open challenge. It is now time to define the operators needed to produce relevant segmentations of natural scenes.

Résumé du chapitre 3

L'objectif de ce chapitre est de donner au lecteur un aperçu des opérateurs morphologiques utiles en traitement d'images. D'une part, nous mettons l'accent sur les opérateurs géodésiques, desquels il est possible de construire des filtres de nivellement puissants, en se basant sur des principes de reconstruction d'image. La géodésie est également au cœur du calcul de la Ligne de Partage des Eaux qui, au moyen de marqueurs judicieusement choisis, permet de contrôler la segmentation d'images en fonction de besoins spécifiques. D'autre part, nous nous intéressons aux fonctions distance pouvant être attribuées à un masque binaire. Nous montrons que ces fonctions distance permettent de résoudre des problèmes de segmentation subjective jouant un rôle important dans l'interpolation de fonctions.

Une bonne compréhension de ces opérateurs ainsi que des techniques de filtrage et de segmentation présentées dans ce chapitre permettra de bien appréhender les méthodes proposées dans la deuxième partie de cette dissertation.

Chapter 3

Morphological Image Processing

Mathematical morphology provides an original framework for image processing and features extraction. Too often, image processing is considered as a straightforward extension of signal processing, which mainly focuses on the analysis and filtering of frequencies. While the latter is of theoretical and practical importance, it does not include the analysis and processing of shapes. Morphological image processing aims to address that omission.

This chapter is an introduction to morphological image processing. In section 3.1, we introduce the reader to dilation and erosion operators from which compounds as well as residual and geodesic operators are constructed. The next sections focus on particular applications of geodesy. Image simplification and the process of flooding and razing a topographic surface are presented in section 3.2. The watershed transformation, which is obtained via a particular kind of flooding, is presented in section 3.3 and constitutes the pivot of morphological image segmentation. Finally, generalised distance functions and their use within regularised segmentations are discussed in section 3.4.

All the concepts presented in this chapter play an essential role in our depth estimation methodology. The concluding remarks on this chapter provide necessary insights into the way these morphological operators will be applied to solving correspondence problems in the field of stereo.

3.1 Operators

In the following section, we present some of the most fundamental operators in mathematical morphology. For the sake of simplicity, we shall consider in the first instance, that $\mathbf{I}$ is a binary image of width W and height H, i.e. $\mathbf{I} : W \times H \rightarrow \{0, 1\}$. We denote by $S(\mathbf{I})$, the set of pixels being activated in the binary mask of $\mathbf{I}$, i.e. $S(\mathbf{I}) = \{(x, y) \mid \mathbf{I}[x, y] = 1\}$. The morphological operators introduced here are driven by what is called a *structuring element* $\mathbf{B}$. It is defined as a set of points $S(\mathbf{B}) = \{(d_x, d_y)\}$ characterising an arbitrary shape and is centred at the origin.

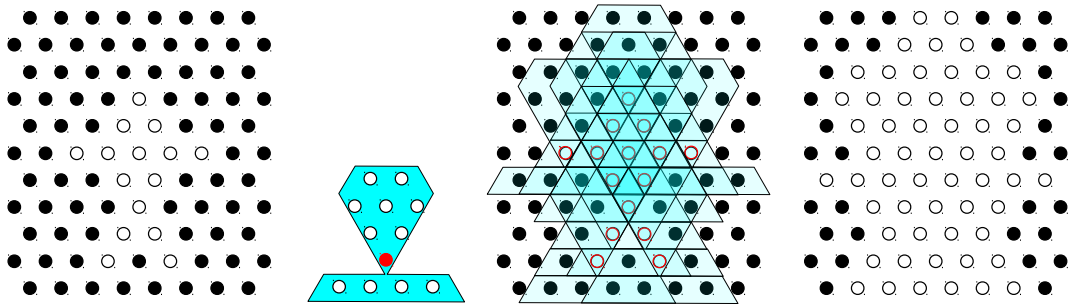


FIGURE 3.1: Illustration of the dilation process. The points of (*a*) the input binary image $\mathbf{I}$ are shown as white discs when they belong to $S(\mathbf{I})$, black discs otherwise. (*b*) The structuring element controlling the dilation is, in this example, a set of twelve points, comprising its origin highlighted in red. (*c*) The structuring element is shifted several times across the image plane, so that its origin superposes with each point belonging to $S(\mathbf{I})$. The union of these shifted structuring elements yields (*d*) the output of the dilation.

Dilation

The binary dilation operator $\delta_{\mathbf{B}}$ controlled by structuring element $\mathbf{B}$, takes as input a binary image $\mathbf{I}$. It consists of the formation of a binary image $\delta_{\mathbf{B}}(\mathbf{I})$, such that each point $(x, y) \in S(\mathbf{I})$ transforms into the structuring element $\mathbf{B}$, shifted so that its centre superposes with (x, y). The union of all these shifts of $\mathbf{B}$ eventually yields $\delta_{\mathbf{B}}(\mathbf{I})$. The points belonging to such a binary dilation of $\mathbf{I}$ are defined as

$$S(\delta_{\mathbf{B}}(\mathbf{I})) = \bigcup_{(d_x, d_y) \in S(\mathbf{B})} \{(x + d_x, y + d_y) \mid (x, y) \in S(\mathbf{I})\}$$

From the above equation, we notice that $\mathbf{B}$ enumerates all the translations of $\mathbf{I}$ of which the union constitutes the product of the dilation. The dilation of $\mathbf{I}$ then reduces to equation 3.1.

$$\delta_{\mathbf{B}}(\mathbf{I}) : (x, y) \mapsto \sup_{(d_x, d_y) \in S(\mathbf{B})} \mathbf{I}[x - d_x, y - d_y] \tag{3.1}$$

Erosion

Compared to the dilation, the erosion operator has exactly the opposite effect on a binary shape. Let $\varepsilon_{\mathbf{B}}(\mathbf{I})$ be the binary erosion of image $\mathbf{I}$ under structuring element $\mathbf{B}$. Any shift of $\mathbf{B}$ reduces itself to at most a single point, being the centre of the shifted structuring element. A shift of $\mathbf{B}$ transforms into a point if and only if every point belonging to the shift of $\mathbf{B}$ lies in $S(\mathbf{I})$. In other

words, the binary erosion $\varepsilon_{\mathbf{B}}(\mathbf{I})$ is defined as

$$S(\varepsilon_{\mathbf{B}}(\mathbf{I})) = \bigcap_{(d_x,d_y)\in S(\mathbf{B})} \{(x,y) \mid (x+d_x, x+d_y) \in S(\mathbf{I})\}$$

In the same way as the dilation of an image has been expressed by equation 3.1 as a supremum of translated images, the above relation leads to equation 3.2 for the erosion.

$$\varepsilon_{\mathbf{B}}(\mathbf{I}) : (x,y) \mapsto \inf_{(d_x,d_y)\in S(\mathbf{B})} \mathbf{I}[x+d_x, y+d_y] \tag{3.2}$$

Dilations and erosions are associated with a wealth of properties, some of which depend on the choice of the structuring element. For example, *if and only if* $S(\mathbf{B})$ contains the centre point $(d_x, d_y) = (0,0)$, then the dilation is extensive, i.e. $S(\mathbf{I}) \subseteq S(\delta_{\mathbf{B}}(\mathbf{I}))$ whilst the erosion is anti-extensive, i.e. $S(\varepsilon_{\mathbf{B}}(\mathbf{I})) \subseteq S(\mathbf{I})$. The reader will find a comprehensive treatment of morphological operator properties in [Serra, 1983] and [Meyer, 1979].

Compounds

Dilations and erosions serve as the main ingredients for building compound morphological operators. We describe three which are standard in mathematical morphology and used in our work.

Dilation and erosion sequences The dilation of size $\lambda \in \mathbb{N}^+$ under structuring element $\mathbf{B}$ is written $\delta_{\lambda\mathbf{B}}$. It is composed of a sequence of dilations applied to the input image.

$$\delta_{\lambda\mathbf{B}}(\mathbf{I}) = \underbrace{\delta_{\mathbf{B}} \circ \ldots \circ \delta_{\mathbf{B}}}_{\lambda \text{ times}}(\mathbf{I}) \tag{3.3}$$

In the same way, the erosion of size $\lambda \in \mathbb{N}^+$, written as $\varepsilon_{\lambda\mathbf{B}}$, may be defined as a succession of erosions of elementary size. These thick dilations and erosions play a major role in the computation of thick gradients, which will be described later in this section.

Opening The opening $\gamma_{\mathbf{B}}$ of an image is the composition of an erosion, followed by a dilation, as defined by equation 3.4.

$$\gamma_{\mathbf{B}}(\mathbf{I}) = \delta_{\mathbf{B}}(\varepsilon_{\mathbf{B}}(\mathbf{I})) \tag{3.4}$$

Closing The closing $\varphi_{\mathbf{B}}$ of an image is the composition of a dilation, followed by an erosion, as defined by equation 3.5.

$$\varphi_{\mathbf{B}}(\mathbf{I}) = \varepsilon_{\mathbf{B}}(\delta_{\mathbf{B}}(\mathbf{I})) \tag{3.5}$$

Greyscale image processing

It is common in greyscale morphology to interpret an image $\mathbf{I}$ composed of N distinct grey levels, as a piling of N binary images $\{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$ representing the level sets of $\mathbf{I}$, such that $\mathbf{I}_i[x, y] = 1 \Leftrightarrow \mathbf{I}[x, y] \geqslant i$. All of these operators may therefore be applied to each level set, meaning equations 3.1 to 3.5 remain valid for greyscale image processing. The structuring element $\mathbf{B}$, as originally defined, is then said to be *flat*, since it operates independently on each level set of image $\mathbf{I}$.

3.1.1 Residues and operators

A residue is defined as the subtraction of an image $\mathbf{I}'$ from an image $\mathbf{I}$, such that $\mathbf{I}[x, y] - \mathbf{I}'[x, y] \geqslant 0$ for every pixel (x, y) being part of the image domain. The most employed residues in mathematical morphology include the following:

The morphological gradient The morphological gradient [Rivest et al., 1993] of an image $\mathbf{I}$ is defined as the difference between the products of its dilation and its erosion. Using an isotropic structuring element $\mathbf{H}$ of unitary size, the morphological gradient of image $\mathbf{I}$ is defined as:

$$\|\nabla \mathbf{I}\| = \delta_{\mathbf{H}}(\mathbf{I}) - \varepsilon_{\mathbf{H}}(\mathbf{I}) \tag{3.6}$$

It is of course possible to evaluate the gradient values with respect to a particular direction, by choosing a structuring element which points only towards the chosen direction. Alternatively, one could decide to compute a thick gradient, by increasing the size of the structuring element.

The top-hat Using a homothetic structuring element $\mathbf{H}$, denoted as $\lambda\mathbf{H}$ for $\lambda \in \mathbb{N}$, the top-hat transformation enables the extraction of crests and peaks which have a width at least equal to 2λ pixels with respect to the image lightness function. The top-hat is defined by equation 3.7.

$$\mathrm{TH}_{\lambda}(\mathbf{I}) = \mathbf{I} - \gamma_{\lambda\mathbf{H}}(\mathbf{I}) \tag{3.7}$$

A variant of this transformation enables the extraction of valleys and holes, and is referred to as the *black top-hat*. The latter is defined as $\varphi_{\lambda\mathbf{H}}(\mathbf{I}) - \mathbf{I}$.

Residual operators

Residual operators provide a means of aggregating residues computed across different scales; the scale relating to the size of the structuring element in use. A residual operator is defined by the combination of two operators denoted as $\diamondsuit^{(1)}$ and $\diamondsuit^{(2)}$. Both operators need to verify that $\diamondsuit^{(1)}_{\lambda}(\mathbf{I}) - \diamondsuit^{(2)}_{\lambda}(\mathbf{I})$ is always a residue, whatever the scale λ is. The residual operator yields the union or the supremum of the residues discovered at each scale, depending on whether the

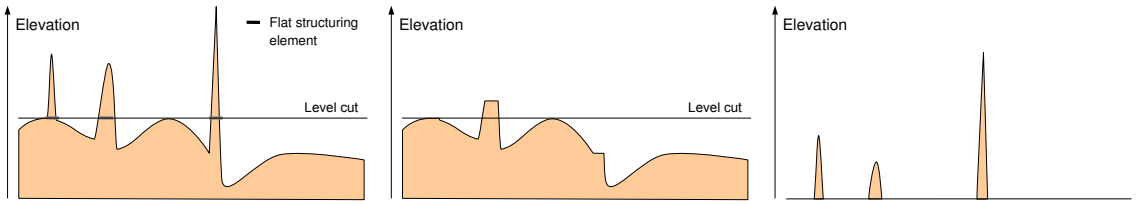


FIGURE 3.2: The white top-hat operator. (*a*) One-dimensional function with multiple elevation levels. The structuring element is represented here as a small segment and its origin is assumed to be the centre of the segment. It operates on the level sets of the function. At the location of the highlighted level cut, we can see that only one of the three peaks can include the structuring element. (*b*) At each level set, the opening suppresses the connected components which cannot include the structuring element. (*c*) The subtraction of this opening from the initial function yields to the white top-hat of this function.

Algorithm 3.1 Template for multi-scale residues computation

function COMPUTEULTIMATERESIDUES($\mathbf{I}$, $\diamondsuit^{(1)}$, $\diamondsuit^{(2)}$)
  Initialise λ to 0 ▷ scale
  Initialise $\mathbf{I}_{\mathtt{ult}}$ to an image filled with 0, having the size of $\mathbf{I}$ ▷ ultimate residues
  Initialise $\mathbf{I}_{\mathtt{arg}}$ to an image filled with 0, having the size of $\mathbf{I}$ ▷ scale of ultimate residues
  Initialise `convergence` to `False`
  while `convergence` $=$ `False` **do**
    $\mathbf{I}_{\mathtt{res}} \leftarrow \diamondsuit_{\lambda}^{(1)}(\mathbf{I}) - \diamondsuit_{\lambda}^{(2)}(\mathbf{I})$
    UPDATEARGUMENT($\mathbf{I}_{\mathtt{res}}$, $\mathbf{I}_{\mathtt{ult}}$, $\mathbf{I}_{\mathtt{arg}}$, λ)
    $\mathbf{I}_{\mathtt{ult}} \leftarrow \sup(\mathbf{I}_{\mathtt{res}}, \mathbf{I}_{\mathtt{ult}})$
    $\lambda \leftarrow \lambda + 1$
    if $(\diamondsuit_{\lambda}^{(1)}(\mathbf{I}) = \diamondsuit_{\lambda-1}^{(1)}(\mathbf{I}))$ **then**
      `convergence` $=$ `True`
  return $\mathbf{I}_{\mathtt{ult}}$ and $\mathbf{I}_{\mathtt{arg}}$

function UPDATEARGUMENT($\mathbf{I}_{\mathtt{res}}$, $\mathbf{I}_{\mathtt{ult}}$, $\mathbf{I}_{\mathtt{arg}}$, λ)
  for all (x, y) belonging to the image domain **do**
    if $\mathbf{I}_{\mathtt{res}}[x, y] > \mathbf{I}_{\mathtt{ult}}[x, y]$ **then**
      $\mathbf{I}_{\mathtt{arg}}[x, y] \leftarrow \lambda + 1$

input is a binary or a greyscale image. An argument function that records the scale at which the residue has been discovered, generally accompanies the output. Algorithm 3.1 gives a full template enabling the computation of these multi-scale residues. The reader will find a thorough treatment of numerical residues in [Beucher, 2005].

Example: distance function of a binary image Let $\mathcal{D}$ be the distance function computed over the binary image $\mathbf{I}$. The distance $\mathcal{D}[x,y]$ at a given point (x,y) of the image plane corresponds to the length of the shortest path originating from any point $(x_0,y_0) \notin S(\mathbf{I})$ and ending at point (x,y). This implies that $\mathcal{D}[x,y] = 0 \Leftrightarrow (x,y) \notin S(\mathbf{I})$. The argument function accompanying the output of the following residual operator

$$\begin{aligned} \diamondsuit^{(1)} &: \lambda \rightarrow \diamondsuit_\lambda^{(1)} = \varepsilon_{\lambda\mathbf{H}} \\ \diamondsuit^{(2)} &: \lambda \rightarrow \diamondsuit_\lambda^{(2)} = \varepsilon_{(\lambda+1)\mathbf{H}} \end{aligned}$$

constitutes the desired distance function $\mathcal{D}$. It should be noted that any erosion of $\mathbf{I}$ can be expressed from the distance function itself, i.e. $S(\varepsilon_{\lambda\mathbf{H}}(\mathbf{I})) = \{(x,y) \,|\, \mathcal{D}[x,y] \geqslant \lambda\}$. It is also possible to obtain a similar relation for the dilation: let $\bar{\mathcal{D}}$ be the distance function of the binary image $1 - \mathbf{I}$. Then the dilation of $\mathbf{I}$ may be expressed as $S(\delta_{\lambda\mathbf{H}}(\mathbf{I})) = \left\{(x,y) \,|\, \bar{\mathcal{D}}[x,y] \leqslant \lambda\right\}$.

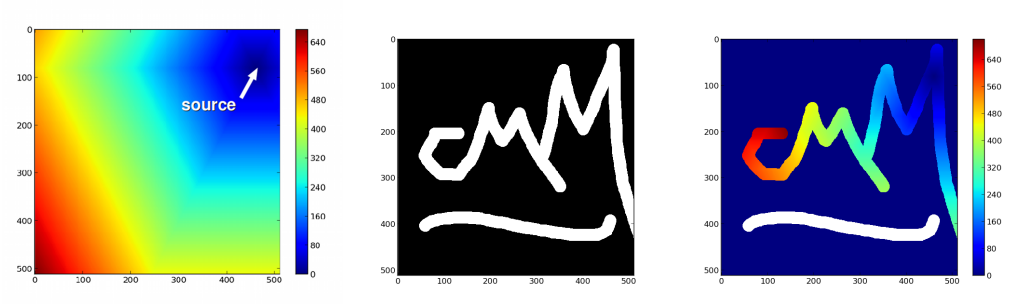
3.1.2 Geodesic operators

In the above example, the shortest path from (x_0,y_0) to a point (x,y) could have been defined by the segment joining these two points. Geodesic operators are designed to control the freedom to connect two points using any arbitrary path across the whole image domain. They achieve this by means of a *mask function*.

In binary morphology, the mask function is represented by a binary image, denoted as $\mathbf{\Omega}$. A *geodesic path* between two image points (x_i,y_i) and (x_j,y_j) within mask $\mathbf{\Omega}$ is a sequence of points $\{(x_k,y_k)\}_{k=0}^{\ell}$ of arbitrary length ℓ, such that:

- the first and final points of the sequence correspond to (x_i,y_i) and (x_j,y_j) respectively
- (x_k,y_k) is a neighbour of (x_{k+1},y_{k+1}), for any $0 \leqslant k < \ell$
- $(x_k,y_k) \in S(\mathbf{\Omega})$, for all $0 \leqslant k \leqslant \ell$

The geodesic distance $\mathcal{D}_{\mathbf{\Omega}}[x,y]$ controlled by the mask function $\mathbf{\Omega}$ and computed for the binary image $\mathbf{I}$, corresponds to the length of the shortest path, amongst all those satisfying the following properties: the path must originate from a point $(x_0,y_0) \,|\, (x_0,y_0) \notin S(\mathbf{I}) \wedge (x_0,y_0) \in S(\mathbf{\Omega})$, the path must end at (x,y), and all points of the path must be included in $\mathbf{\Omega}$. As illustrated by figure 3.3, such a path may not exist, in which case $\mathcal{D}_{\mathbf{\Omega}}[x,y] = +\infty$. This geodesic distance function leads to the definition of geodesic operators, comprising the application of a threshold



(*a*) (*b*) (*c*)

FIGURE 3.3: Distance functions. (*a*) The distance function computed using successive erosions controlled by an elementary hexagonal structuring element. The erosions have been applied to a binary image containing just one hole, the location of which is indicated by the white arrow. (*b*) The mask $\boldsymbol{\Omega}$. Its introduction enables us to restrict the domain into which morphological operators proceed. (*c*) The resulting geodesic distance function from the very same source point. Notice that the distance is infinite for one of the connected components appearing in $\boldsymbol{\Omega}$ since there is no path connecting the source point to any of its points.

on the geodesic distance values, in a similar manner to that shown in the previous section. The function to which a geodesic operator is applied is called the *marker function* and should always be included in the mask function.

Binary geodesic dilation

Computing the geodesic dilation of the marker function $\mathbf{I}$ at scale λ and within mask $\boldsymbol{\Omega}$ is equivalent to computing the geodesic distance function of the inverted image $\mathbf{I}$, up to distance λ, within the domain defined by $\boldsymbol{\Omega}$. This can be achieved by applying successive elementary dilations to image $\mathbf{I}$ (in place of the erosions on $1-\mathbf{I}$) which should never lie external to the mask function. The geodesic dilation of marker $\mathbf{I}$ inside mask $\boldsymbol{\Omega}$ and at scale λ, is expressed by equation 3.8.

$$\mathrm{D}^{\lambda}_{\boldsymbol{\Omega}}(\mathbf{I}) = \underbrace{\delta\left(\ldots\left(\underbrace{\delta\left(\underbrace{\delta(\mathbf{I})\cap\boldsymbol{\Omega}}_{\text{Scale 1}}\right)\cap\boldsymbol{\Omega}}_{\text{Scale 2}}\right)\ldots\right)\cap\boldsymbol{\Omega}}_{\text{Scale }\lambda} \tag{3.8}$$

Binary geodesic reconstruction

The scale at which the geodesic dilation of a marker function $\mathbf{I}$ inside the mask function $\boldsymbol{\Omega}$ reaches idempotence, corresponds to the scale at which marker $\mathbf{I}$ has *reconstructed* mask $\boldsymbol{\Omega}$. Let $\mathrm{R}_{\boldsymbol{\Omega}}(\mathbf{I})$ be the *geodesic reconstruction* of mask $\boldsymbol{\Omega}$ by the marker function $\mathbf{I}$. $\mathrm{R}_{\boldsymbol{\Omega}}(\mathbf{I})$ is expressed by

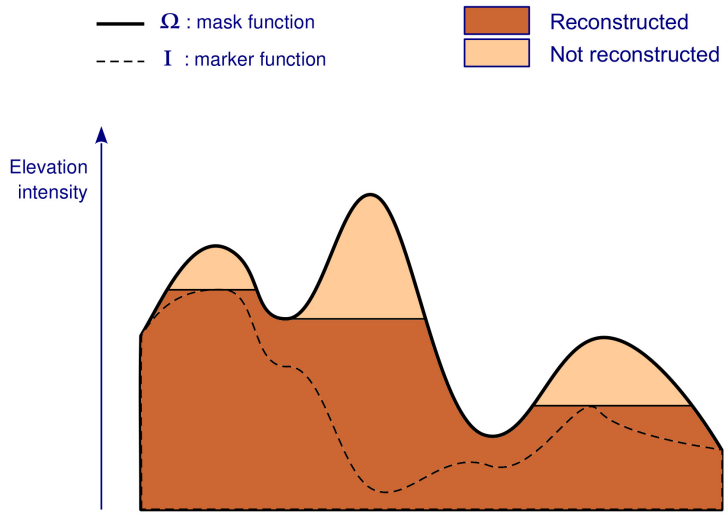


FIGURE 3.4: Geodesic reconstruction of a mask function Ω by a marker image $\mathbf{I}$

relation 3.9.

$$\mathrm{R}_{\mathbf{\Omega}}(\mathbf{I}) = \mathrm{D}^{\lambda^\star}_{\mathbf{\Omega}}(\mathbf{I}) \mid \mathrm{D}^{\lambda^\star}_{\mathbf{\Omega}}(\mathbf{I}) = \mathrm{D}^{\lambda^\star+1}_{\mathbf{\Omega}}(\mathbf{I}) \tag{3.9}$$

In fact, the geodesic reconstruction of binary mask $\mathbf{\Omega}$ restores the connected components of $\mathbf{\Omega}$, marked by $\mathbf{I}$. We say that two image points (x_i, y_i) and (x_j, y_j) belong to the same connected component of $\mathbf{\Omega}$ if and only if there exists a geodesic path between them in $\mathbf{\Omega}$. Therefore, in order to reconstruct a particular connected component of $\mathbf{\Omega}$, at least one of its points must belong to $S(\mathbf{I})$.

Geodesic reconstruction of greyscale images

In the same way as we remarked about the standard morphological operators, a binary geodesic operator may be applied to each level set of the greyscale marker image, in conjunction with the corresponding level set of the mask image, to supply the product of its equivalent numerical operator.

In greyscale image processing, the elementary and the thick geodesic dilations for greyscale images are defined by relations 3.10 and 3.11 respectively, where the marker and mask functions are defined such that $\mathbf{I}[x, y] \leqslant \mathbf{\Omega}[x, y]$ across the entire image domain.

$$\mathrm{D}_{\mathbf{\Omega}}(\mathbf{I}) = \inf\{\delta(\mathbf{I}), \mathbf{\Omega}\} \tag{3.10}$$

$$\mathrm{D}^{\lambda}_{\mathbf{\Omega}}(\mathbf{I}) = \underbrace{\mathrm{D}_{\mathbf{\Omega}} \circ \ldots \circ \mathrm{D}_{\mathbf{\Omega}}}_{\lambda \text{ times}}(\mathbf{I}) \tag{3.11}$$

The expression of the geodesic reconstruction operator R in equation 3.9 remains unchanged.

Symbol	Description
Structuring elements	
B	a structuring element
H	an isotropic structuring element of elementary size
Morphological transformations	
$\delta_B(f)$	Dilation of f:
	$\delta_B(f)[x] = \sup_{x' \in B} f[x - x']$
$\varepsilon_B(f)$	Erosion of f:
	$\varepsilon_B(f)[x] = \inf_{x' \in B} f[x + x']$
$D_g^1(f)$	Geodesic dilation of f inside mask g:
	$D_g^1(f) = \inf\{\delta_H(f), g\}$
$R_g(f)$	Geodesic reconstruction of g from marker function $f \leqslant g$:
	$R_g(f) = D_g^{+\infty}(f) = D_g^1\left(\ldots\left(D_g^1(f)\right)\ldots\right)$
$R_g^*(f)$	Dual geodesic reconstruction of g from marker $f \geqslant g$:
	$R_g^*(f) = -R_{-g}(-f)$
$\gamma_B(f)$	Opening of f:
	$\gamma_B(f) = \delta_B(\varepsilon_B(f))$
$\gamma_B^{(R)}(f)$	Opening by reconstruction of f:
	$\gamma_B^{(R)}(f) = R_f(\varepsilon_B(f))$

TABLE 3.1: Notation for mathematical morphology. It is assumed that f and g both constitute arbitrary functions, such that x is an antecedent of that function.

Furthermore, a *dual geodesic reconstruction* is also defined, when $\mathbf{I}[x,y] \geqslant \mathbf{\Omega}[x,y]$ for any point (x,y) of the image domain. It corresponds to the additive inverse of the geodesic reconstruction of $(-\mathbf{I})$ inside $(-\mathbf{\Omega})$ and is written $R_{\mathbf{\Omega}}^*(\mathbf{I}) = -R_{-\mathbf{\Omega}}(-\mathbf{I})$.

Reconstruction operators on greyscale images have a wide scope of application ranging from image filtering and simplification to hierarchical segmentation. Some of these will be described in the next sections of this chapter.

Further reading

By referring to [Beucher, 1990], the reader will find an authoritative treatment of geodesic operators in mathematical morphology. It should be noted that the implementation of reconstruction operators by successive geodesic dilations is far from optimal computationally. Fast algorithms based on priority queues [Beucher and Beucher, 2011] are currently available in good computer libraries of mathematical morphology. Finally, it is important not to confuse the dual reconstruction with the succession of geodesic erosions of a marker function inside a particular mask function. Several definitions exist for the geodesic erosion, as discussed in [Beucher, 2011].

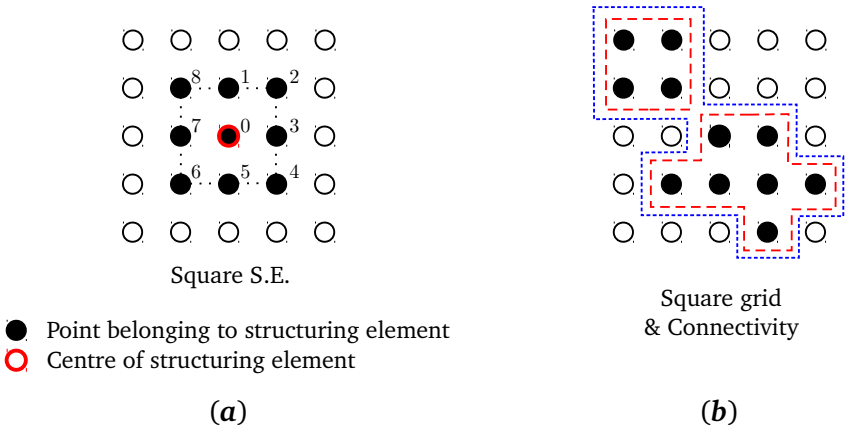


FIGURE 3.5: The square grid. (*a*) The isotropic structuring element is represented by a square composed of nine points (including the centre). In contrast to the hexagonal grid, the distance between any two neighbour points varies. (*b*) The square grid is subject to an ambiguity regarding the interpretation of connectivity. Either *all* points coloured in black or *all* points coloured in white are connected. One can immediately appreciate the danger of that ambiguity, with respect to the reconstruction of connected components.

3.1.3 Image processing grids

The morphological operators which we have just presented, will be used throughout the rest of this work. Before investigating their applications, let us provide some insights about how to use them with mathematical morphology libraries for 2D image processing.

It is first necessary to define the grid onto which the image will be processed. This grid is usually either square or hexagonal. In the square grid, a point either shares the eight neighbours illustrated in figure 3.5(*a*), or a subset of those in the directions 1, 3, 5 and 7 of the square grid. The reason for defining two different kinds of connectivities becomes clearer when looking at figure 3.5(*b*). We can see that it is impossible to ascertain whether the pixels coloured black form a single connected component, in which case those coloured white would form two separate connected components; or if the contrary is true. To solve that ambiguity, it is possible to consider the 8-connectivity for the white pixels and the 4-connectivity for the black pixels, or vice-versa.

In general, it is preferable to resort to the hexagonal grid illustrated in figure 3.6, which is not subject to these problems. A point in the hexagonal grid shares six equidistant neighbours. The hexagonal structuring element of elementary size is then defined as a set of seven points, which includes the centre, whilst directional structuring elements are segments defined over one of the six available directions, as shown in figure 3.6(*b*).

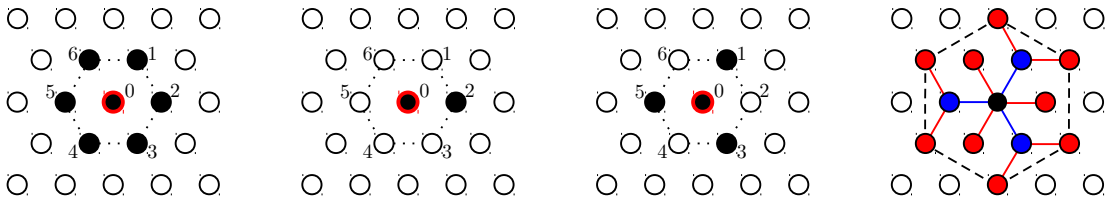


FIGURE 3.6: The hexagonal grid and hexagonal structuring elements of unitary size. (*a*) The isotropic structuring element is represented by a hexagonal shape composed of seven points (including the centre), (*b*) A directional structuring element is represented by a segment composed of the centre point plus one of the six available directions, (*c*) A tripod structuring element is composed of three directions, each separated from one another by an angle of 120°, (*d*) A conjugated hexagon.

Towards more isotropic structuring elements

When operating on a hexagonal grid using the unitary hexagon as structuring element, the dilation of a point of size λ produces a hexagon of size λ. In order to generate further isotropic dilations, the hexagons can be replaced by dodecagonal structuring elements. Unfortunately, there is no dodecagon of elementary size defined for the hexagonal grid. [Beucher, 2012] shows that it is nonetheless possible to construct dodecagons of size $\lambda = \lambda_1 + 2\lambda_2$ from the composition of a dilation based on the hexagonal structuring element of size λ_1, and a dilation characterised by a *conjugated* hexagonal structuring element of size $2\lambda_2$. Conjugated hexagons are obtained by combining the two *tripods* available for the hexagonal grid, as shown in figure 3.6(*c*), and are consequently always of an even size. According to [Beucher, 2012], a hexagon of size $\lambda_1^\star = \sqrt{3}(2-\sqrt{3})\lambda$ and a conjugated hexagon of size $2\lambda_2^\star$, where $\lambda_2^\star = (2-\sqrt{3})\lambda$, would be necessary to construct a dodecagon of size λ from a single point. However, $\lambda_1^\star$ and $\lambda_2^\star$ are not valid dimensions on a discrete grid. For this reason, λ_1 is chosen as either the floor or the ceiling of $\lambda_1^\star$ so that λ_1 and λ share the same parity. λ_2 is finally deduced as $\lambda_2 = \frac{1}{2}(\lambda - \lambda_1)$.

In section 3.1.1, we saw that it is useful to define thick dilations (or erosions) in a recursive way, in particular when generating distance functions. As we progressively increment the value of λ, we observe an increment in λ_2 every time and only when λ_1 is decremented. Therefore, if we limit ourselves to using hexagons and their conjugates as structuring elements, we need to remember the two dilations of thickness $\lambda - 1$ and λ to deduce the dodecagonal dilation of size $\lambda + 1$, which renders the iterative process somewhat cumbersome. However, we notice that as long as λ_1 is incremented, it is simple to derive the dilation state from the preceding one, by using the hexagonal dilation, as figure 3.7 confirms. Therefore, the problem amounts to investigating whether a particular operator allows us to deduce the dodecagonal dilation at scales where both

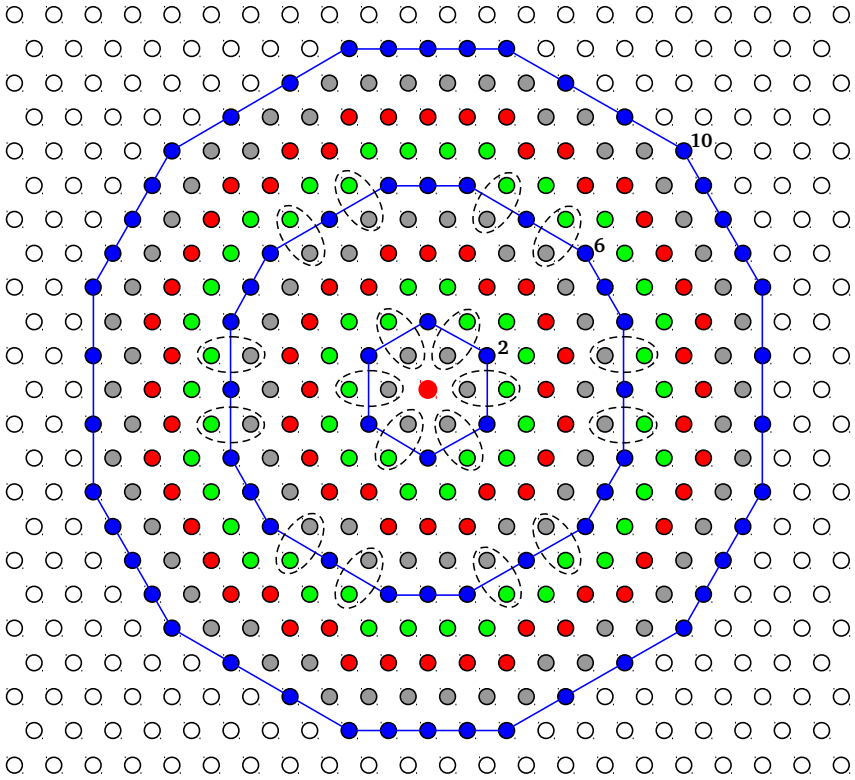


FIGURE 3.7: Dodecagonal dilations. The partial dilation takes place at scales $\lambda = 2$, $\lambda = 6$, $\lambda = 10$, etc.

λ_1 and λ_2 are decremented and incremented respectively.

The answer lies in the *partial dilation*, in which the conjugate hexagon of size 2 is derived from the regular hexagon of size 1. A point belongs to the partial dilation of a binary image $\mathbf{I}$ if and only if it appears simultaneously in the two dilations of $\mathbf{I}$, each obtained using a distinct tripod defined for the hexagonal grid. With regard to figure 3.7, the green points contained in the ellipses with dotted outlines are precisely those which do not belong to the partial dilation of the shape obtained after iteration steps $\lambda = 1$ and $\lambda = 5$. We conclude that in order to produce dodecagonal dilations in a recursive fashion, it is sufficient to replace the standard hexagonal dilation with a partial dilation every time we observe an increment in λ_2.

In short

The consistency between the connectivities of foreground and background pixels is essential when processing images which use morphological operators, hence the choice of the hexagonal grid. For some applications, such as the computation of distance functions, the availability of isotropic structuring elements is also key in order to limit the bias effects resulting from dilations or erosions across a discrete grid. Unfortunately, isotropic structuring elements of elementary size are not usually defined on the grid, and they therefore necessitate special constructs, like the one presented for computing dodecagonal dilations.

3.2 Image simplification and levelling

The reconstruction operators discussed in section 3.1.2 have several uses in image simplification and filtering. One of the most common examples is the opening by reconstruction defined in table 3.1. Unlike the classical opening, the reconstruction step restores the flat zones of the greyscale function which have not disappeared after the erosion, and does not alter the shape of the contours which were visible in the original image. With respect to this work, the aim of image simplification is to facilitate the segmentation of a scene, which makes this latter characteristic particularly desirable.

Sequential alternate levelling

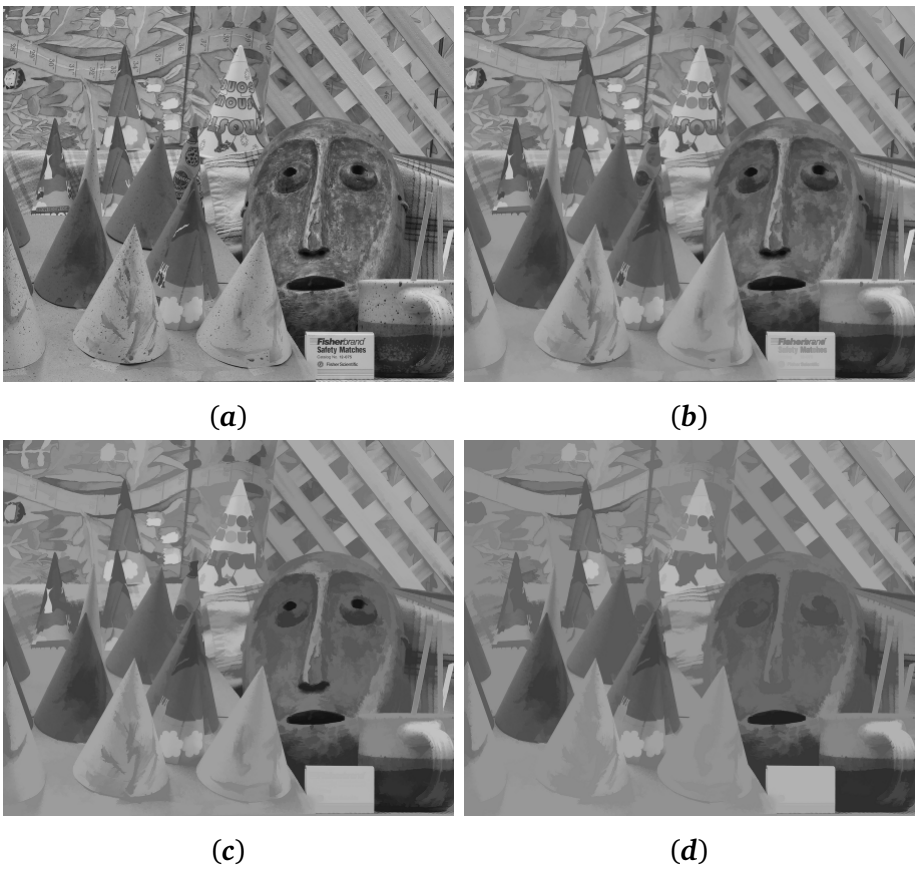
The geodesic reconstruction and dual reconstruction operators may be used to respectively *raze* and *flood* the topographic surface represented by a greyscale image function $\mathbf{I}$. The parts of the surface undergoing one of these two phenomena then become flat. The razed parts always lie under the original surface, whilst the flooded parts cover the original surface.

The composition of a razing and a flooding yields a *levelling*. Let $\mathrm{Rz}(\mathbf{I}, \mathbf{M}) = \mathtt{R}_{\mathbf{I}}(\mathbf{M})$ be the razing of $\mathbf{I}$ controlled by marker $\mathbf{M} \leqslant \mathbf{I}$ and $\mathrm{Fl}(\mathbf{I}, \mathbf{M}) = \mathtt{R}^{*}_{\mathbf{I}}(\mathbf{M})$, the flooding of image $\mathbf{I}$ controlled by marker $\mathbf{M} \geqslant \mathbf{I}$. The sequential alternate levelling of an image $\mathbf{I}$ up to scale λ, is defined by relation 3.12, where the razing and the flooding order can be swapped according to the user's discretion.

$$\underbrace{\mathrm{Rz}\left(\mathrm{Fl}\left(\ldots \underbrace{\mathrm{Rz}\left(\mathrm{Fl}\left(\underbrace{\mathrm{Rz}\left(\mathrm{Fl}\left(\mathbf{I}, \delta\mathbf{I}\right), \varepsilon\mathbf{I}\right)}_{\text{Scale 1}}, \delta_2\mathbf{I}\right), \varepsilon_2\mathbf{I}\right)}_{\text{Scale 2}} \ldots, \delta_\lambda\mathbf{I}\right), \varepsilon_\lambda\mathbf{I}\right)}_{\text{Scale } \lambda} \tag{3.12}$$

We can observe that all flat zones in the current image which do not resist an opening or closing operation at a given scale are filtered out. In a similar fashion to the opening and the closing by reconstruction, a flat zone which has not been entirely destroyed by an erosion or a dilation is reconstructed, which prevents the occurrence of new and false contours. The fact that the sequential alternate levelling is a scale-adaptive filter enables us to choose the scale at which the filtering of flat zones should be effective, and, as a result, to employ that within the computation of a multi-scale gradient (cf. section 4.1).

Image levelling is not restricted to sequential alternate levelling. It is possible to use any other marker function to generate the required images. The reader will find a careful treatment of levelling theory and practice in [Meyer, 2004]. Just as with any morphological filters, levelling is prone to the generation of false colours when applied independently to each channel of a colour



(a) (b)

(c) (d)

FIGURE 3.8: A pyramid of alternate sequential levelling. (a) Input image, (b) Output at levelling scale $\lambda = 5$. Small components, such as the dots on the cones are filtered out and the texture across the scene's mask is flattened. As the strength of the levelling increases, larger components disappear from the output, as can be seen for (c) scale $\lambda = 10$ and (d) scale $\lambda = 20$.

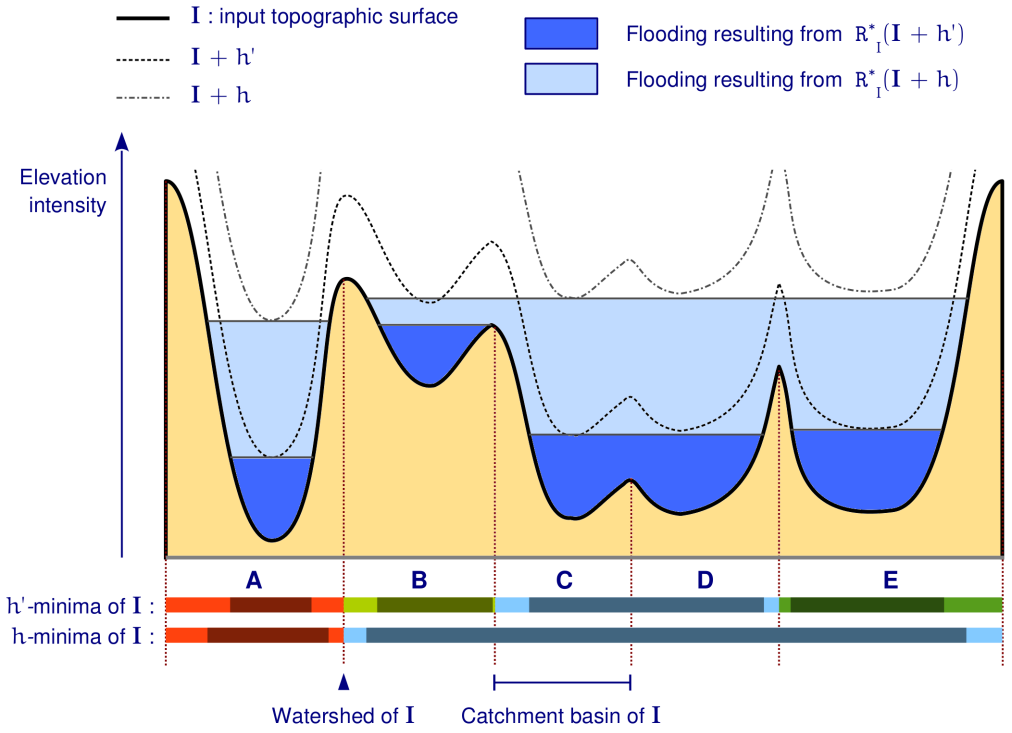


FIGURE 3.9: Synchronous flooding based on the depth of lakes. The dual reconstruction of $\mathbf{I}+h'$ over the initial topographic surface $\mathbf{I}$ is a flooding of $\mathbf{I}$. The resulting lakes are shown in dark blue. Their maximum depth is either equal to h' (cf. catchment basins A, C, D and E) or inferior to h' in case of absorption (cf. catchment basin B). The h'-minima then correspond to the set of locations where the original topographic surface has been flooded. We notice that when $h > h'$, the h'-minima of $\mathbf{I}$ are contained in the h-minima of $\mathbf{I}$. This property of inclusiveness is useful when it comes to generating hierarchical segmentations of an image from the h-minima of its gradient.

image. Several methods exist to improve the levelling quality of colour imagery as detailed in [Zanoguera Tous, 2001]. Amongst these, the autarkic vectorial levelling approach proposed in [Gomila, 2001] preserves the colours of the marker and the image functions.

Synchronous flooding and h-minima

So far, we have seen that flooding a topographic surface amounts to performing a dual geodesic reconstruction. We will now focus on a particular type of flooding. Consider the dual reconstruction of $\mathbf{I}$ by the marker function $\mathbf{I}+h$, for any $h \in \mathbb{N}^+$, as illustrated in figure 3.9. This flooding ensures that all resulting lakes reach a depth equal to or less than h. Lakes which cannot reach the maximum depth h are absorbed by neighbour lakes. The h-minima of a discrete function $\mathbf{I}$ are represented by a binary image $\mathbf{M}_h(\mathbf{I})$ verifying relation 3.13.

$$S(\mathbf{M}_h(\mathbf{I})) = \{(x,y) \mid \mathrm{R}^*_{\mathbf{I}}(\mathbf{I}+h)[x,y] > \mathbf{I}[x,y]\} \tag{3.13}$$

h-minima are very useful in the context of image segmentation. They can be computed on the image gradient to extract the image zones which satisfy a certain homogeneity. Small gradient peaks are indeed covered by the h-minima, for a sufficiently large elevation h, while the image points which do not belong to the h-minima delineate the original image locations where the contrast is sharply defined. Furthermore, h-minima satisfy the property that $S(\mathbf{M}_h(\mathbf{I})) \supseteq S(\mathbf{M}_{h'}(\mathbf{I}))$ for any $h > h'$, as established in [Beucher, 2013b]. This latter is particularly useful as it induces a segmentation hierarchy based on the instant, i.e. the depth, at which two neighbouring lakes merge.

3.3 Markers-driven segmentation

Segmenting an image involves partitioning the latter into a set of disjoint connected components which together comprise the full image domain. These connected components are often referred to as *regions* or *cells*. The objective of image segmentation depends on the target application. For instance, if we are interested in segmenting out the homogeneous regions of a scene, then the partition must be composed of cells which satisfy the desired homogeneous criterion. We might also be interested in segmenting an image into at least two regions: the *foreground* which constitutes an object of interest and the *background* representing the rest of the image plane.

The ability to localise the areas of interest is convenient for controlling the product of a segmentation algorithm. As far as the detection of homogeneous regions is concerned, the h-minima introduced in section 3.2 are likely to constitute good *markers* of homogeneous areas when extracted from the image gradient. In morphological image processing, the prime tool for segmenting images is the *watershed transformation* [Beucher, 1990, Beucher and Meyer, 1992], an operator which is capable of being driven by image markers.

The watershed transformation

A marker is a connected component defined over the image domain and is systematically assigned a *label*. The label can either identify the connected component or it can convey semantic information, such as the type of object designated by the marker. All markers are represented by a label map, denoted by $\mathcal{L}$, which has the same dimensions as the image being segmented. If (x, y) is contained in a marker, then $\mathcal{L}[x, y] = \ell$, where $\ell > 0$ is the label of that particular marker. If (x, y) is not contained in a marker, then $\mathcal{L}[x, y] = 0$.

Let $\mathbf{S}$ be an image representing a topographical surface, i.e. $\mathbf{S}[x, y]$ encodes the altitude of the surface point that projects onto pixel (x, y). It is possible to compute the watershed of $\mathbf{S}$ *controlled* by the markers represented by the label map $\mathcal{L}$. The computation is achieved by a *uniform* flooding of the topographic surface from the proposed markers. It works as follows:

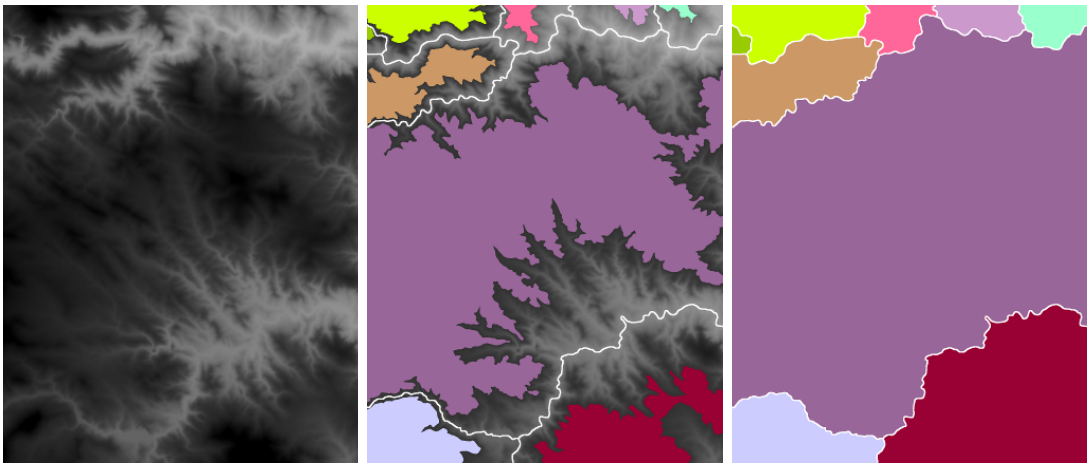
- The number of flooding operations is equal to the number of elevations in $\mathbf{S}$. The operations are performed in ascending order with respect to the altitude and progressively extend the label map $\mathcal{L}$ by means of the flooding results.
- At a given altitude h, the binary mask $\mathbf{\Omega}_h$ is extracted, such that $S(\mathbf{\Omega}_h) = \{(x, y) \mid \mathbf{S}[x, y] \leqslant h\}$. The labelled lakes contained in $\mathcal{L}$ propagate inside $\mathbf{\Omega}_h$ in an isotropic fashion. The label map $\mathcal{L}$ is updated at each propagation step. However the propagation of a label to a point is only effective, if and only if that point has not been allocated a label prior to the previous propagation step. When two or more lakes *with different labels* meet, the junction point becomes part of the final watershed.
- At the end of the process, there is no point in the label map $\mathcal{L}$ which has not been assigned a strictly positive label. Moreover, the labelling provided by the initial markers remains intact. In order to obtain the watershed, it is of course necessary to retain another image which provides a record of the points where lakes of different labels meet.

It is worth noting that the watershed transformation is a blend of *extended* binary geodesic reconstructions, in the sense that the labelling information of the originating markers remains intact. In this type of geodesic reconstruction, when two or more lakes with different labels meet, the points at which they meet lie at an equal geodesic distance from the initial markers. They are said to belong to the *geodesic skeleton of the influence zones* of these markers.

Efficient implementations of the watershed driven by markers are available using priority queues [Beucher and Beucher, 2011], where the priority refers to the elevation value of the topographic function $\mathbf{S}$.

Segmentation in practice

Figure 3.10 shows an example of the watershed transformation on a topographic surface. In practice, we need to search for the topographic surface which, once flooded by the appropriate markers, yields the desired segmentation. In order to segment an image into homogeneous components, the morphological gradient generally suffices as long as the input image is devoid of noise, is in focus, and fully exploits the available dynamic range. Several researches have focused on improving the gradient computed from colour images. [Angulo López, 2003] proposed a raft of experiments, which combine the gradients of the hue, luminance and saturation channels, and in his work, [Risson, 2001] showed how clearly the gradient based on perceptual colour differences improves the quality of segmentation hierarchies. That same gradient was employed in [Hanbury and Marcotegui, 2006] in a controlled fashion, in order to determine the probability of a pixel, which conforms to a particular colour difference in its neighbourhood, delineating the frontier between two objects of interest. The probability map was then used as the topographic surface on which to perform the watershed transformation.



(a) (b) (c)

FIGURE 3.10: The watershed of a topographic surface driven by markers. (a) Image S of 1344×1736 pixels encoding the topographic surface of interest. The greyscale value assigned to a particular pixel is proportional to the altitude of the actual relief and lies between 0 and 255. (b) Visualisation of the chosen markers over the topographical surface. They have been obtained by computing the h-minima of the relief, for an elevation value of $\mathrm{h} = 55$. Using the regular reconstruction operators, only those touching the image border have been retained. Finally, an opening by reconstruction of size 20 pixels has been applied to eliminate markers which are too small. (c) The final catchment basins obtained after the segmentation preserve the labels of their respective markers. They constitute the final state of the label map $\mathcal{L}$. The watershed line is displayed in white.

Unfortunately, the gradient does not always reveal the region contours *explicitly*. Imagine, for example, of a binary image containing a circle with a dotted outline from which you aim to recover the complete circle. A foreground marker is placed inside the circle, while the background marker delineates the image border. By construction of the segmentation, the flooding induced by the watershed can lead one of the markers to leak into the opposite region, by exploiting the contour's leakages, which would thereby prevent the extraction of the target circle. An initial response to that problem, which is generalised to greyscale images, is the *viscous* watershed [Vachier and Meyer, 2005]. In this approach, the level sets of the gradient undergo a series of *closings* in which the size decreases as the elevation increases. As a result, the contours delineated by the regular watershed associated with the updated gradient, remain accurate where the gradient takes high elevation values and elsewhere become more regularised. The result, though, remains sensitive to the parametrisation of the closing sequence. Earlier research in binary image segmentation focused on subjective contours, and essentially relied on *distance functions* [Vincent and Dougherty, 1994], the most famous example being the separation of coffee beans. The next and last section of this chapter proposes a further investigation into these distance functions, but this time generalised to greyscale imagery.

3.4 Generalised distance functions and segmentation

In section 3.1.1, we provided the definition of a distance function $\mathcal{D}$ computed from a binary mask $\mathbf{M}$, which states that, for a given point (x, y) of the image plane, $\mathcal{D}[x, y]$ corresponds to the distance of the shortest path from any point *in* $S(\mathbf{M})$, leading to the point of coordinates (x, y). Each pixel along the path had the same weight to the measured length of the path. We are now interested in adding a constraint to the computation of such distance functions. This time, it is possible that each pixel may contribute differently to the measured length of the path under consideration. This means that, when computing the distance function, several iterations may be necessary to move from one point to a neighbour point. The number of iterations required to perform such a move is determined by the topographic surface $\mathbf{I}$. In this section, we present an algorithm for generating generalised distance functions as well as some applications to segmentation.

Construction

Algorithm 3.2 provides a generic framework for the computation of generalised distances. Let $\mathcal{D}$ be the sought distance function. Initially, $\mathcal{D}[x, y] = 0$ for all $(x, y) \in S(\mathbf{M})$ and is set to $+\infty$ otherwise. At any time t of the iteration process, the binary mask $\mathbf{M}_t$ containing the points which have already been reached from $S(\mathbf{M})$, is defined such that $(x, y) \in S(\mathbf{M}_t) \Leftrightarrow \mathcal{D}[x, y] \neq +\infty$. Now, in order to find which points to visit next, we need to dilate $\mathbf{M}_t$ using an isotropic structuring element. In the binary mask distance computation, all newly discovered points are *immediately* allocated a distance equal to $t + 1$. In the generalised distance computation, a *delay*, encoded by the delay function $\mathbf{C}$, indicates the *maximum* number of iterations remaining before a point lying in $\delta(\mathbf{M}_t)$ is eventually reached by one of its already discovered neighbours belonging to $\mathbf{M}_t$. The delay function is therefore decremented at every iteration in line 10 and furthermore can never be increased again, as is demonstrated by line 12. When the delay $\mathbf{C}[x, y]$ eventually reaches zero, $\mathcal{D}[x, y]$ is updated with the current iteration time in line 15 remaining unchanged in subsequent iterations. Finally, in line 11, (x, y) will belong to $\mathbf{M}_t$ only after it has been visited.

The implementation of the FETCHCLOCKUPDATE routine in line 12 depends mainly on the target application. Algorithms 3.3 and 3.4 present two particularly useful study cases:

- The generalised distance computed using the clock update mechanism of algorithm 3.3 corresponds to a true geodesic distance on the relief $\mathbf{I}$. Indeed, the propagation delay between two neighbouring points is determined in function of their difference in altitude with respect to the topographic surface $\mathbf{I}$. It is of course essential that the measured altitude and a unitary displacement on the image plane share the same units. The user of algorithm 3.2 can adapt the factor α for that particular purpose.

Algorithm 3.2 Geodesic distance from a binary image on a topographic surface

function COMPUTEGEODESICDISTANCE($\mathbf{I}$, $\mathbf{M}$, α)
  Initialise image $\mathcal{D}$ with the size of $\mathbf{I}$, all pixels set to $+\infty$ ▷ Distance function
  Initialise image $\mathbf{C}$ with the size of $\mathbf{I}$, all pixels set to $+\infty$ ▷ Delay function

  for all (x, y) belonging to the image domain **do**
    if $\mathbf{M}[x, y] = 1$ **then**
      $\mathcal{D}[x, y] \leftarrow 0$
      $\mathbf{C}[x, y] \leftarrow 0$

  $t \leftarrow 0$ ▷ Iteration time
  while $\exists\,(x, y) \mid \mathbf{C}[x, y] > 0$ **do**
    $\mathbf{C} \leftarrow \max(0,\ \mathbf{C} - 1)$
    $\mathbf{M}_t \leftarrow$ FETCHCURRENTMASK($\mathcal{D}$)
    $\mathbf{C} \leftarrow \min\{\mathbf{C}, \alpha \times$ FETCHCLOCKUPDATE($\mathbf{I}, \mathbf{M}_t$)$\}$

    $t \leftarrow t + 1$
    for all $(x, y) \mid \mathbf{C}[x, y] = 0$ **do**
      $\mathcal{D}[x, y] \leftarrow \min\{\mathcal{D}[x, y],\ t\}$

  return $\mathcal{D}$

function FETCHCURRENTMASK($\mathcal{D}$)
  return binary mask $\mathbf{M}$, such that $\mathbf{M}[x, y] = 1 \Leftrightarrow \mathcal{D}[x, y] \neq +\infty$

Algorithm 3.3 Delay update based on the differences of surface elevation

function FETCHCLOCKUPDATE($\mathbf{I}$, $\mathbf{M}$)
  Initialise η with the size of $\mathbf{I}$, all pixels set to $+\infty$ ▷ Viscosity function
  for all directional structuring elements of elementary size, h **do**
    $\mathbf{I}' \leftarrow \delta_h(\mathbf{I}) - \varepsilon_h(\mathbf{I})$
    $\mathbf{M}' \leftarrow \delta_h(\mathbf{M}) - \varepsilon_h(\mathbf{M})$
    for all (x, y) belonging to the image domain **do**
      if $\mathbf{M}'[x, y] = 0$ **then**
        $\mathbf{I}'[x, y] = +\infty$
    $\eta \leftarrow \min\{\eta, \mathbf{I}'\}$
  return η

Algorithm 3.4 Delay update based on the surface elevation

function FETCHCLOCKUPDATE($\mathbf{I}$, $\mathbf{M}$)
  Initialise η with the size of $\mathbf{I}$, all pixels set to $+\infty$ ▷ Viscosity function
  $\mathbf{M}' \leftarrow \delta_H(\mathbf{M}) - \varepsilon_H(\mathbf{M})$
  for all (x, y) belonging to the image domain **do**
    if $\mathbf{M}'[x, y] = 1$ **then**
      $\eta[x, y] = \mathbf{I}[x, y]$
  return η

- The clock update mechanism of algorithm 3.4 is simpler in that the function $\mathbf{I}$ directly indicates the *viscosity* of the dilations at every pixel of the image plane. In other words, the more the relief is elevated, the slower the dilations become.

Accelerated computation of generalised geodesic distances

The execution speed constitutes the main limitation of algorithm 3.2: the number of iterations required to fulfil the computation of the distance function equals the maximum distance value to be allocated to the distance function. When interested in computing the distance function according to the update rule of algorithm 3.4, one should note that a faster and much simpler algorithm is feasible.

Let $\mathcal{D}_\star$ be the geodesic distance function we seek. $\boldsymbol{\eta}$ represents the viscosity function which assigns each pixel the *positive* delay required to be reached by one of its neighbours. $\mathcal{N}_{(x,y)}$ represents the set of points lying in the direct neighbourhood of point (x,y). The properties of the geodesic distance function $\mathcal{D}_\star$ are summarised as follows:

1. $\mathcal{D}_\star[x,y] = 0$ for all $(x,y) \in \mathbf{M}$
2. $\mathcal{D}_\star[x,y] = \min_{(x',y')\in\mathcal{N}_{(x,y)}}\{\mathcal{D}_\star[x',y'] + \boldsymbol{\eta}[x,y]\}$ for all $(x,y) \notin \mathbf{M}$

Thus the second property states that if a point (x,y) lies outside the initialisation mask $\mathbf{M}$, its shortest geodesic distance to the mask by traversing the viscosity field $\boldsymbol{\eta}$, equals the shortest geodesic distance to the mask $\mathbf{M}$ from a point lying in the direct neighbourhood (x,y), plus the delay it takes for this point to be reached by this neighbour. Combining these two properties, we can write that:

$$\mathcal{D}_\star[x,y] = \min\left\{\mathcal{D}_\star[x,y], \min_{(x',y')\in\mathcal{N}_{(x,y)}}\left\{\mathcal{D}_\star[x',y'] + \boldsymbol{\eta}[x,y]\right\}\right\}$$

$\mathcal{D}_\star[x,y]$ can be found by reformulating the above equation as a recurrence relation. To proceed, we define $\mathcal{D}_0$ as the initial distance function, so that, in the same way as algorithm 3.2, $\mathcal{D}_0[x,y] = 0$ if and only if $(x,y) \in S(\mathbf{M})$, otherwise $\mathcal{D}_0[x,y] = +\infty$. This initialisation ensures that the first property related to the geodesic distance function is satisfied. The recurrence relation is then expressed for $t \in \{1,2,\ldots\}$ as:

$$\begin{aligned}
\mathcal{D}_t[x,y] &= \min\left\{\mathcal{D}_{t-1}[x,y], \min_{(x',y')\in\mathcal{N}_{(x,y)}}\left\{\mathcal{D}_{t-1}[x',y'] + \boldsymbol{\eta}[x,y]\right\}\right\} \\
\Leftrightarrow \mathcal{D}_t[x,y] &= \min\left\{\mathcal{D}_{t-1}[x,y], \min_{(x',y')\in\mathcal{N}_{(x,y)}}\left\{\mathcal{D}_{t-1}[x',y']\right\} + \boldsymbol{\eta}[x,y]\right\}
\end{aligned}$$

By observing that $\varepsilon_{\mathbf{H}}\left(\mathcal{D}_{t-1}\right)[x,y] = \min\left\{\mathcal{D}_{t-1}[x,y], \min_{(x',y')\in\mathcal{N}_{(x,y)}}\{\mathcal{D}_{t-1}[x',y']\}\right\}$, the last

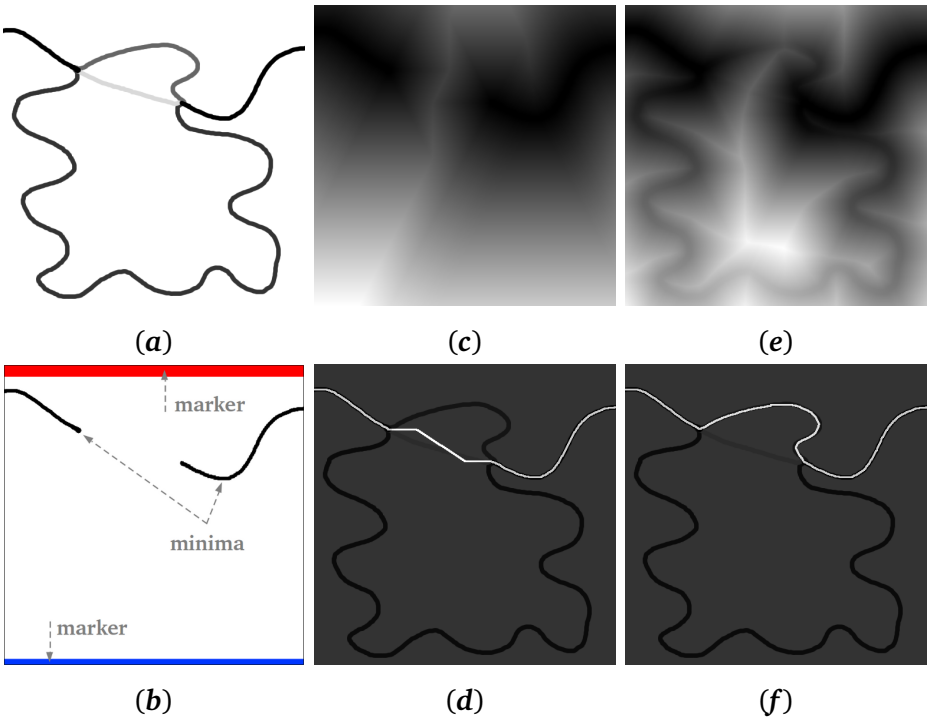


FIGURE 3.11: Comparison of regular and generalised distance functions. (a) Topographic surface $\mathbf{I}$, (b) The minima represent the source points belonging to $S(\mathbf{M})$ within the computation of (c) the regular distance function and (e) the generalised distance function using the clock update of algorithm 3.4. The watershed transform is applied to the inverted distance functions using the foreground and background markers shown in blue and red respectively. This results in (d) the watershed using the regular distance function and (f) the watershed using the generalised distance function.

relation simplifies into:

$$\mathcal{D}_t = \min\{\mathcal{D}_{t-1}, \varepsilon_{\mathbf{H}}(\mathcal{D}_{t-1}) + \eta\} \tag{3.14}$$

We observe that this sequence monotonically decreases until it reaches convergence, since $\mathcal{D}_{t-1} \geqslant \mathcal{D}_t \geqslant 0$. Upon convergence for $t = t^\star$, the second property of the geodesic distance function is satisfied, and we have $\mathcal{D}_{t^\star+1} = \mathcal{D}_{t^\star} = \mathcal{D}_\star$. Finally, we note that, replacing η by a constant, such that $\eta = 1$, the recurrence relation 3.14 enables the computation of the regular distance function, from a binary mask, as seen in section 3.1.1.

Applications to segmentation

Figure 3.11 shows an example where we employ the generalised distance function in order to find a shortest path binding two minima of a topographic surface $\mathbf{I}$. The length of the path is defined as the sum of elevation values of $\mathbf{I}$ for every pixel belonging to the path. The underlying segmentation is based on the watershed transformation addressed in section 3.3 using two markers positioned near the top and bottom image borders, while the topographical surface being flooded corresponds to the inverted distance function. Being able to perform a *guided*

interpolation of an arbitrary function may prove particularly useful within the processing of disparity space images, and we shall go deeper into such distance functions in the second part of the thesis.

Further reading

The reader may be interested in the presentation of generalised distance functions, which were introduced for the first time in mathematical morphology, in [Beucher, 1990]. The proposed algorithm is slightly different, since it enables points of the same plateau to be discovered simultaneously, thus avoiding the usual iteration process. Generalised distances were also used in Beucher's work to adapt the scale of morphological operators across road images, for which the filtering intensity was dependent on the depth of vehicles.

Summary

The purpose of this chapter was to accompany the reader on a guided tour of mathematical morphology. The dilation and erosion operators are at the heart of morphological image processing. On the one hand, we placed particular emphasis on the geodesic operators, from which we derived the image levelling; a filtering method based on geodesic reconstruction principles. Geodesy also proved essential to the watershed transformation which, when combined with markers, allows us to control the segmentation of images according to specific needs. On the other hand, we devoted an important part of this discussion to the computation and usage of distance functions. First, we explained how the distance functions computed from binary masks could be used to express the binary dilations and erosions, both in the classic and geodesic configurations. Then we mentioned some of their applications with respect to the segmentation of images and showed their effectiveness when capturing subjective contours from greyscale images. To a certain extent, distance functions, when used in conjunction with the watershed transformation, provide a powerful means of interpolating functions. We concluded this chapter with the presentation of generalised distance functions, which enable more subtle interpolations in complex scenarios.

An understanding of the operators as well as of the filtering and segmentation techniques presented here, will prove essential to comprehending the methods introduced in the second part of this thesis. With respect to depth map computation, we will be interested in three problems which deal with mathematical morphology. The first is how to generate a segmentation of the scene, which captures all object boundaries with little over-segmentation. The second problem we will address is the generation of equivalent stereo segmentations. The ability to drive the watershed using markers will be particularly useful in that respect. The third and ultimate problem we aim to solve, is the segmentation of the disparity space volume into foreground and

background voxels. To proceed, a distance function based on the cells of the image partitions will be employed in the context of disparity map interpolation.

Part II

Methodology

Résumé du chapitre 4

Ce chapitre détaille notre utilisation de la Ligne de Partage des Eaux dans le but de générer des segmentations qui soient cohérentes pour l'analyse de mises en correspondance entre images stéréoscopiques.

Afin de contrôler une segmentation recourant à la L.P.E., il est nécessaire de choisir non seulement les marqueurs d'objets à segmenter, mais aussi la surface topographique à inonder, de manière appropriée. Concernant les scènes naturelles, la magnitude du gradient de l'image semble être un bon choix de surface topographique. Pourtant, le gradient brut est souvent sujet au bruit, et les contours que nous percevons comme étant saillants dans l'image ne s'y traduisent pas de manière aussi prononcée. Pour cette raison, nous proposons un gradient morphologique régularisé et multi-échelle, qui préserve la dynamique des contours perçus par la vision humaine.

Ensuite, en partant de ce type de surface topographique, nous présentons un ensemble de méthodes d'extraction de marqueurs. L'extraction de marqueurs exploite principalement les inondations dont les principes ont été introduits au chapitre 3. Différents critères, tels que la profondeur des lacs ou l'aire de leur surface, permettent d'interrompre les inondations. La combinaison de ces critères nous permet d'obtenir des segmentations d'objets relativement pertinentes. Ces mêmes critères sont ensuite utilisés pour générer des sur-segmentations d'images, lesquelles seront employées dans notre algorithme de calcul de profondeur le plus avancé. Ces sur-segmentations offrent un bon compromis entre l'aire et la saillance des régions les composant. En effet, plus les contours d'une région sont fortement prononcés dans le gradient de l'image, plus cette région aura des chances d'être segmentée correctement, même si elle est de petite taille. Plus les contours sont associés à un faible gradient, plus la région devra atteindre une taille importante pour être parfaitement segmentée. Les attributs contrôlant les seuils minimaux de contraste et d'aire s'adaptent automatiquement au contenu de l'image afin de garantir que les segmentations obtenues préservent leur pertinence et leur utilité au travers de la base de données considérée.

Les segmentations obtenues par les méthodes proposées dans ce chapitre contrôleront ainsi les algorithmes de calcul de cartes de profondeur présentés dans la suite de l'exposé.

Chapter 4

Controlled Watershed Segmentations

In this work, the segmentation of stereo images always constitutes the starting point for the estimation of disparity maps. As indicated in chapter 2, regions will provide the shape of relevant correlation supports across which the superimpositions of stereo images will be processed and analysed. Therefore, in order for the analysis of image correlations to be pertinent, it is essential that the regions represent the objects composing the scene. Unfortunately, computing a semantic partition for an image of unknown nature remains a difficult task; one which generally necessitates learning and pattern recognition strategies. Rather than address this problem, we propose to design a segmentation algorithm delineating most of the relevant contours based on perceptual cues.

Since the images we are dealing with mostly represent in-door scenes, we aim to compute segmentations which take the perception of contrast into account. These segmentations must be sufficiently coarse to turn regions into meaningful aggregation supports, but should nevertheless capture thin image structures. To fulfil this objective using the watershed transformation, an appropriate topographical surface is needed as input. In section 4.1, we present a morphological and multi-scale gradient which enhances the saliency of the perceived contours across the images. This gradient will be useful as soon as we need to deal with microstereopsis imagery which is not in focus. Next, in section 4.2, we show how to extract meaningful markers from the provided gradient according to three alternatives. The first consists of obtaining the markers by separating the h-minima of the gradient in places where the latter is subject to leakages. In the second and third alternatives, markers directly originate from the minima of an altered gradient. We investigate reconstruction mechanisms enforcing contrast, area and volume criteria on the partition's regions, and the viscous transformation which drastically reduces the number of gradient minima so as to yield pertinent over-segmentations. To conclude, we show in section 4.3 how partitions satisfying different segmentation attributes can be combined to produce the final segmentations.

4.1 Multi-scale enhancement of contour saliency

The morphological gradient described by equation 3.6, which is in fact nothing other than the gradient magnitude of the topographical surface associated with the processed image, provides a natural means of highlighting the high frequencies of an image. However, the contours of interest might not be sufficiently well accentuated and several reasons can account for this fact.

The first problem is that sensor noise is systematically amplified in the gradient. Either it should be filtered out before computation, or adequate algorithms should be employed instead. For example, [Lerallut, 2006] defines the gradient, at a given image point, as the distance between the two most significant modes of the brightness distribution covering the neighbourhood of the pixel under consideration. In the same spirit, [Arbelaez et al., 2011] build a gradient by comparing brightness distributions from either side of a circular patch centred at the pixel of interest and for different orientations. The second problem is therefore the choice of the patch size. Indeed, significant sizes widen the scope of analysis and enable one to be attentive to smoothly evolving brightness transitions. This aspect is particularly useful when having to deal with out-of-focus imagery. Nevertheless, when using large filtering kernels, thin homogeneous areas will be filled with high gradient values, which is not desirable for the detection of contours. The third problem, specific to colour imaging, is the interpretation of colour differences; an interpretation which depends on the type of images being studied. For instance, the CIE colour distances [McLaren, 1976] based on perceptual differences should be the most appropriate for natural scene photography. Further studies of morphological colour gradients for the same context but also for medical imagery have been proposed in [Angulo López, 2003]. However, when dealing with broadcast images, other transformations such as the enhanced HLS transformation [Demarty and Beucher, 1998] may be more suitable for discriminating objects in a scene.

In order to address these three problems, we developed a novel multi-scale and morphological gradient [Bricola et al., 2015], building on the regularised gradient [Beucher, 1990].

4.1.1 Regularised gradients

The regularised gradient $\mathbf{G}$ is a morphological and multi-scale gradient determined for greyscale images. Let $\mathbf{I}$ be a greyscale image. In section 3.1.1, we explained how to compute thick gradients for different sizes of structuring elements. In fact, each size refers to a specific scale. The computation of the regularised gradient between scales λ_s and λ_e, such that $\lambda_s, \lambda_e \in \mathbb{N}$ and $\lambda_e \geqslant \lambda_s > 0$, involves the computation of a series of thin image gradients $\mathbf{G}_{\lambda_s}, \ldots, \mathbf{G}_{\lambda_e}$. Each thin gradient $\mathbf{G}_\lambda$ is deduced from the corresponding thick gradient of size λ according to algorithm 4.1. Finally, the supremum of all these thin gradients constitutes the final regularised gradient, expressed by equation 4.1.

Algorithm 4.1 Regularised gradient at a particular scale

1: **function** COMPUTEGRADIENTATSCALE($\mathbf{I}$, λ)
2: $\mathbf{G}_\lambda \leftarrow \delta_\lambda(\mathbf{I}) - \varepsilon_\lambda(\mathbf{I})$ ▷ Thick gradient capturing transitions at scale λ
3: $\mathbf{G}_\lambda \leftarrow \mathbf{G}_\lambda - \gamma_\lambda(\mathbf{G}_\lambda)$ ▷ White top-hat discarding contours of thickness $\geqslant \lambda$
4: $\mathbf{G}_\lambda \leftarrow \varepsilon_{\lambda-1}(\mathbf{G}_\lambda)$ ▷ Erosion bringing contours back to original resolution
5: **return** $\mathbf{G}_\lambda$

Algorithm 4.2 Enhanced regularised gradient at a particular scale

1: **function** COMPUTEGRADIENTATSCALE($\mathbf{I}$, λ)
2: $\mathbf{I}_F$ obtained by a strong levelling of $\mathbf{I}$ up to scale λ ▷ cf. equation 3.12
3: $\mathbf{M}_\lambda$ is a binary mask highlighting intensity transition points in $\mathbf{I}_F$ ▷ cf. equation 4.3
4: $\mathbf{G}_\lambda \leftarrow \delta_\lambda(\mathbf{I}_F) - \varepsilon_\lambda(\mathbf{I}_F)$
5: $\mathbf{G}_\lambda \leftarrow \mathbf{G}_\lambda - \gamma_{2\lambda-1}(\mathbf{G}_\lambda)$
6: $\mathbf{G}_\lambda \leftarrow \mathbf{G}_\lambda \times \mathbf{M}_\lambda$
7: **return** $\mathbf{G}_\lambda$

$$\mathbf{G} = \sup_\lambda \{\mathbf{G}_\lambda\} \mid \lambda \in \{\lambda_s, \lambda_s + 1, \ldots, \lambda_e\} \tag{4.1}$$

Assuming that both the dilations and the erosions use isotropic structuring elements, the thick gradient at line 2 captures brightness transitions taking at most 2λ pixels. Relevant contours should also reach a thickness of 2λ pixels. However, it may be that the contours of an object having a thickness smaller than λ pixels have fused. The white top-hat operator at line 3 aims to remove those fusions, since the resulting contours exceed the expected thickness of 2λ pixels. Finally, the erosion in line 4 restores the residual contours to their original resolution.

Thick gradient enhancement

Despite the fact that brightness variations are well captured by the thick gradient, the product of the regularised gradient, computed using algorithm 4.1, does not reflect this quality. In fact, the thick gradient is, by definition, highly sensitive to noise. This means that regions affected by sensor noise will acquire significant gradient values and thus, the top-hat operator which is applied next, will break the dynamic of the crests of interest, as shown in figure 4.1(***d***). It is therefore essential to filter the image before applying the thick gradient operator. In section 3.2, we discussed the way that levelling simplifies images without altering contours. We also explained that a strong levelling of size λ broadly removes structures which cannot be reconstructed after a closing or an opening of size λ. Therefore, the strong levelling seems a very good choice of image pre-filtering, which can be adapted to the desired scale: not only will sharp contours remain intact, but most artefacts caused by the fusion of contours at the given scale will be prevented. Figure 4.2 illustrates how the latter property contributes to a better preservation of thick contours at high scales. Once the gradients are de-noised, there is nothing to prevent the employment of a white top-hat of a greater thickness. Since we are interested only in eliminating contours which

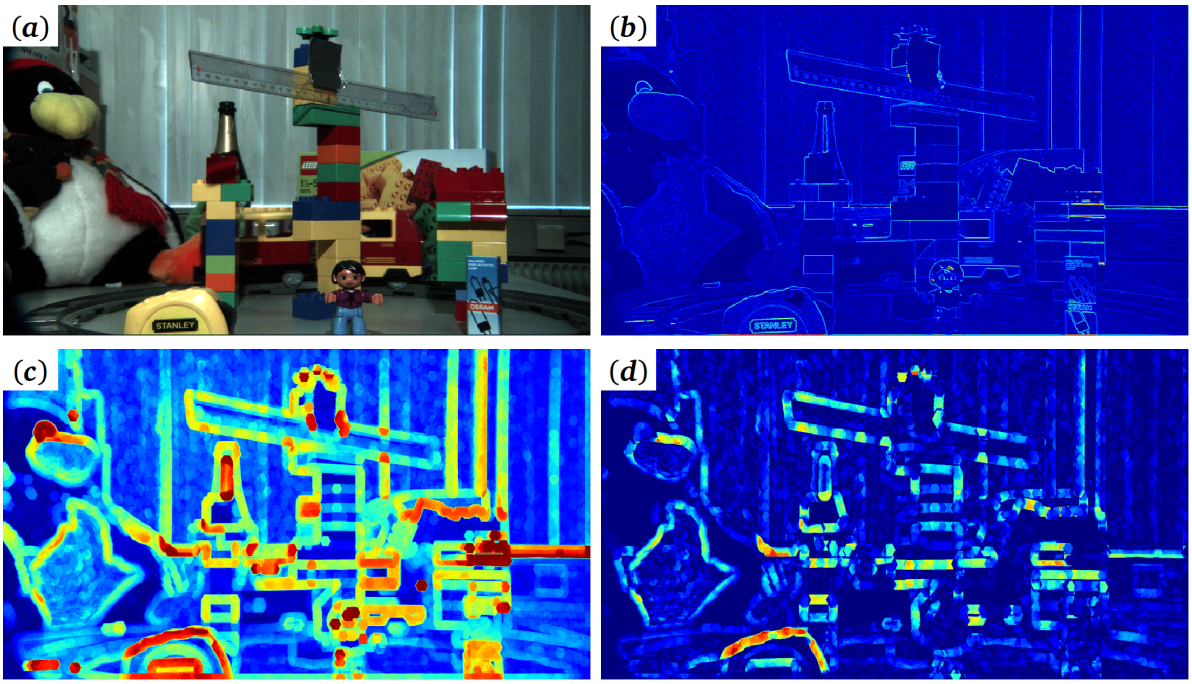


FIGURE 4.1: Increasing the gradient's thickness in algorithm 4.1. (*a*) Input image of size 1920×1080 pixels. (*b*) The morphological gradient of the green channel struggles to capture smooth transitions occurring between the penguin's foot and stomach, the railway lines, the train's windows, etc. (*c*) The thick gradient of size $\lambda = 16$ accentuates these smooth transitions but is severely affected by noise which leads to (*d*) the destruction of the contrast dynamic after having applied the white top-hat operator.

have fused, and have therefore attained the thickness of 4λ pixels, we can adjust the size of the white top-hat to $2\lambda - 1$, as shown in algorithm 4.2, line 5.

Original scale restoration enhancement

In a binary image composed of black and white connected components, the erosion step of algorithm 4.1 correctly restores the residual thick gradient contours to their original resolution. However, in greyscale images displaying smooth brightness transitions, the thick erosion is likely to diminish values captured by the residual thick gradient, as illustrated in figure 4.3. This is the reason our preferred enhanced algorithm proposes a mechanism which does not alter the gradient values.

At each scale, a binary mask highlighting the image locations where a brightness transition occurs, is superimposed over the residual thick gradient. The structure of this mask is dependent on the levelled image $\mathbf{I}_F$ obtained at the scale we are considering, λ. First, we compute the thick morphological average $\mathbf{I}_A$ associated with image $\mathbf{I}_F$ and to scale λ using equation 4.2.

$$\mathbf{I}_A = \frac{1}{2}\left(\delta_\lambda\left(\mathbf{I}_F\right) + \varepsilon_\lambda\left(\mathbf{I}_F\right)\right) \tag{4.2}$$

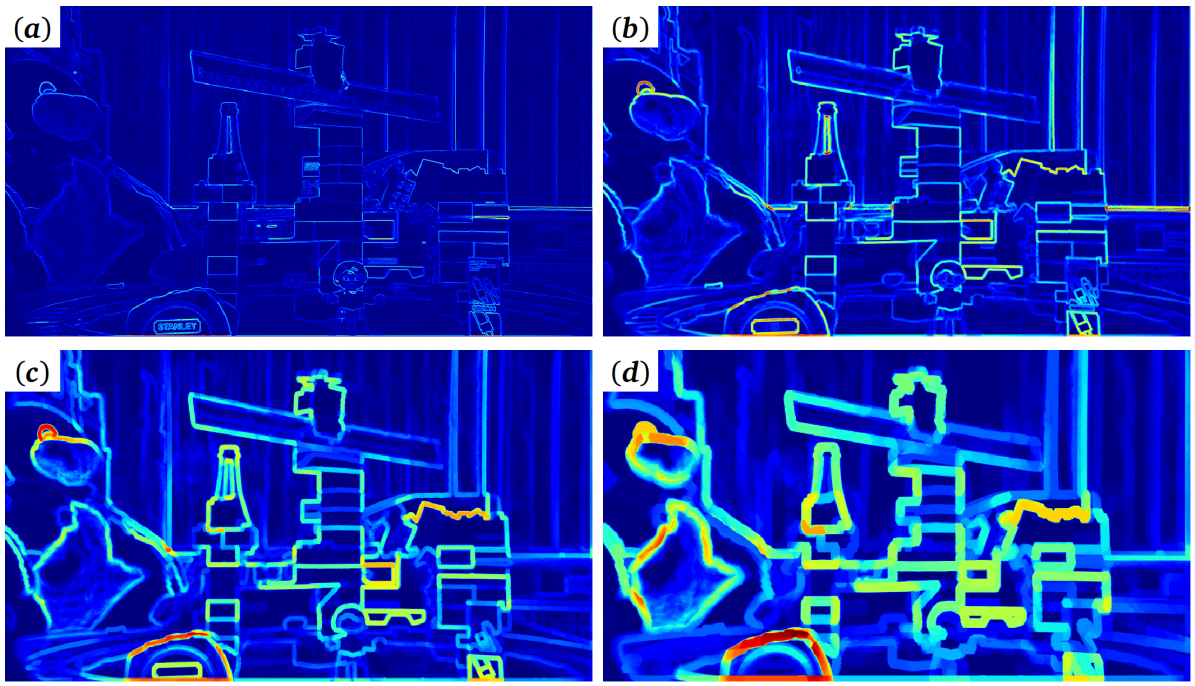


FIGURE 4.2: The pyramid of thick gradients after applying alternate sequential levelling operations in line with the gradient thickness of (a) $\lambda = 1$ pixel, (b) $\lambda = 4$ pixels, (c) $\lambda = 8$ pixels and (d) $\lambda = 16$ pixels, to the green channel. Compared to the original thick gradients, the noise has been filtered out and small structures responsible for a great number of undesirable contour fusions have already been discarded.

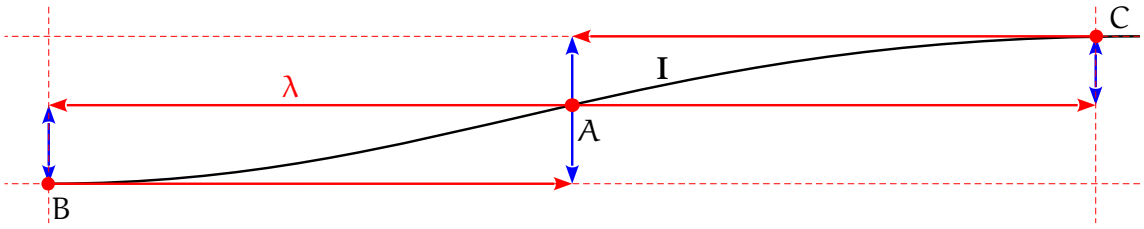


FIGURE 4.3: A limitation induced by the erosion occurring in algorithm 4.1, step 4. The black curve represents a smoothly evolving intensity function I. The blue segments represent the difference of intensity values observed around points A, B and C in a neighbourhood of λ pixels. From this illustration, we can predict that the erosion of size $\lambda - 1$ will, at point A, reduce the thick gradient intensity by a factor of two.

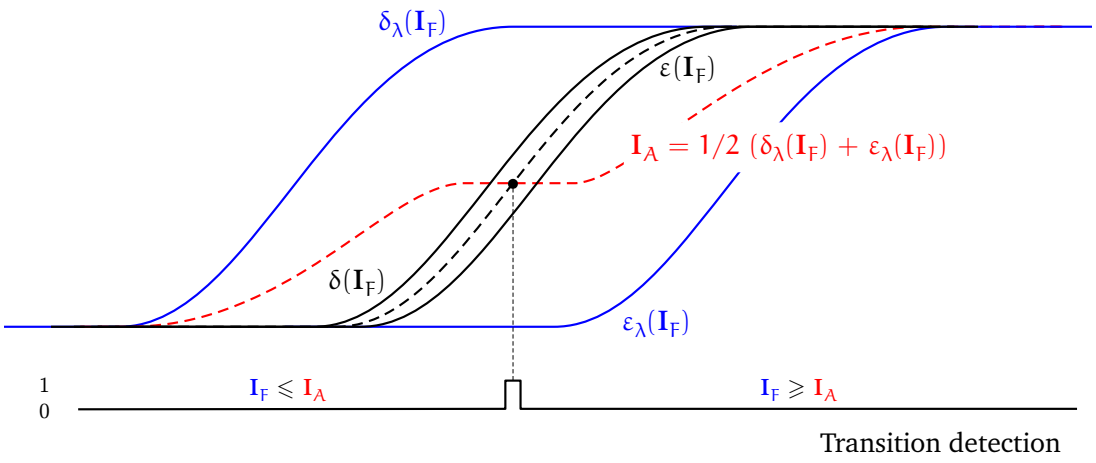


FIGURE 4.4: Relation between the brightness function shown as the black dotted curve, and its morphological average displayed as the red dotted curve. The ideal transition point should be at the intersection of $\mathbf{I}_F$ and $\mathbf{I}_A$, when the function $\mathbf{I}_F$ is not flat.

If we interpret $\mathbf{I}_A$ and $\mathbf{I}_F$ as brightness functions, we can expect that the points where both functions are equal with non-null derivatives will be at the centre of a brightness transition, as illustrated in figure 4.4. Therefore, it is tempting to search for perfect equality between $\mathbf{I}_A$ and $\mathbf{I}_F$. However, this does not always work well in practice, because nothing guarantees that this equality will be effectively found within a discretised space. We can instead, search for points lying in the direct neighbourhood of the intersection of the functions. Figure 4.5 illustrates the procedure enabling such a detection. The points belonging to the mask $\mathbf{M}_\lambda$ of algorithm 4.2 are ultimately defined by the set:

$$S(\mathbf{M}_\lambda) = \{(x,y) \mid \varepsilon\,(\mathbf{I}_F)\,[x,y] < \mathbf{I}_A[x,y] < \delta\,(\mathbf{I}_F)\,[x,y]\} \tag{4.3}$$

where ε and δ denote the isotropic erosion and dilation operators of unitary size, respectively. Note that, by construction, points lying near the end of the brightness transitions might belong to the detections. But since their gradient values are close to zero, the impact on the final regularised gradient is minimal.

Preliminary results

We now have all necessary ingredients to implement algorithm 4.2. Figure 4.6 shows some outputs produced by this algorithm on defocused greyscale images. As expected, contours situated in blurred image areas are more salient in the enhanced regularised gradient than in the morphological gradient. Besides, the levelling of size 1 also clears the gradient of the high frequencies caused by texture, leading to more pertinent catchment basins as can be observed for the image showing coffee beans. The next section is devoted to the integration of colour

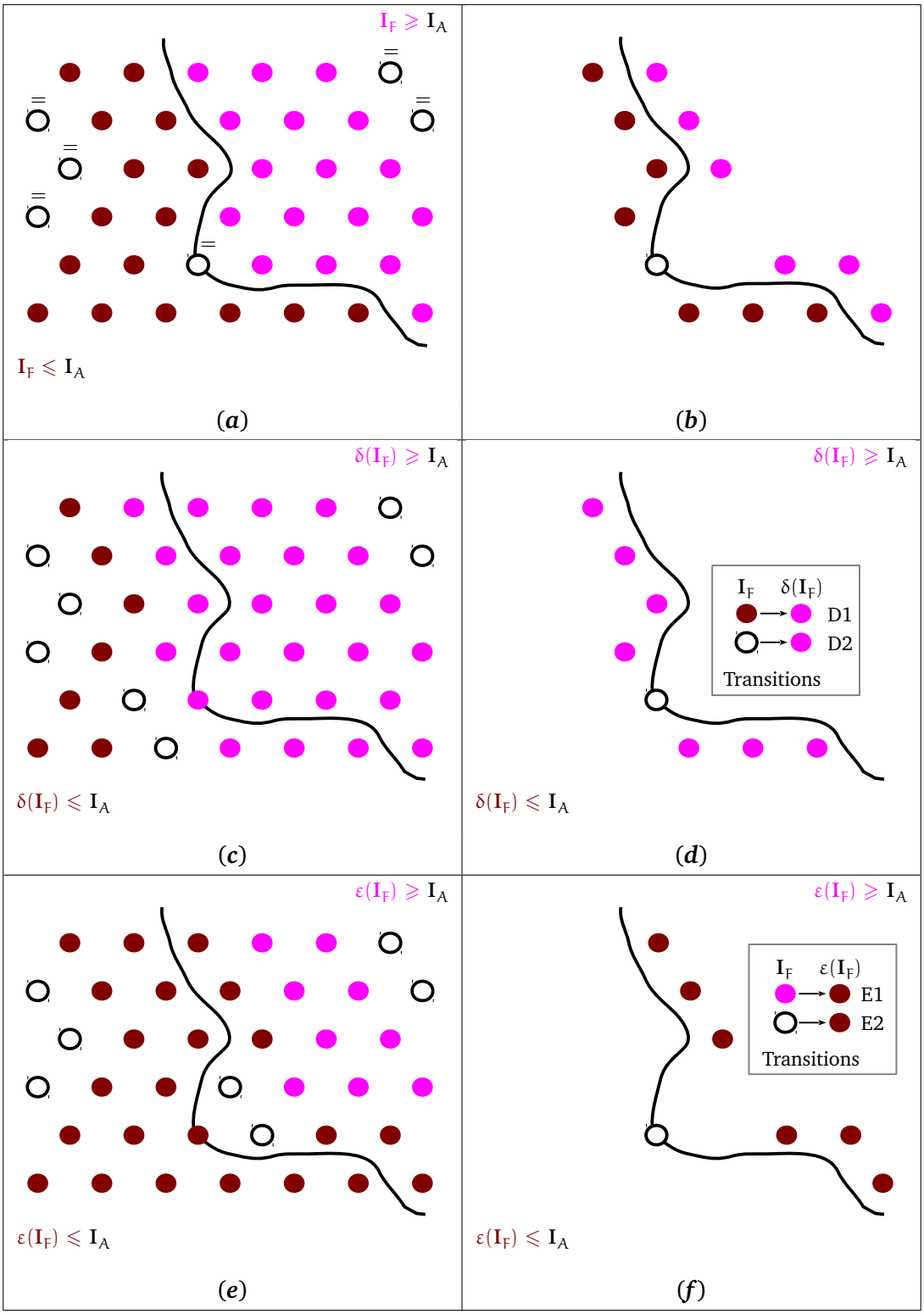


FIGURE 4.5: Detection of the brightness transition points. (*a*) Pixels of I_F having a brightness inferior to I_A are coloured red. Those having a brightness above the morphological average are coloured purple and those of equal brightness are highlighted in white. The same process is repeated by replacing I_F with $\delta(I_F)$ in (*c*) and $\varepsilon(I_F)$ in (*e*). The change of labels between scenarios (*a*) and (*c*) yields the detection of points lying on the intersection of I_F and I_A, or near that intersection with a brightness value just below that found in $\delta(I_F)$. A similar comment can be formulated for (*f*), by comparing scenarios (*a*) and (*e*). The union of the detections in (*d*) and (*f*) yields the transition points in (*b*).

information, which is valuable when it comes to distinguishing objects of different hues and saturations. As a concluding remark regarding regularised gradients, it is worth observing that the argument accompanying the optimisation expressed by equation 4.1 would indicate at which scale a contour with the maximum gradient value has been detected. If we were able to relate the focus strength to the depth of a scene, this argument could encode interesting contour depth cues.

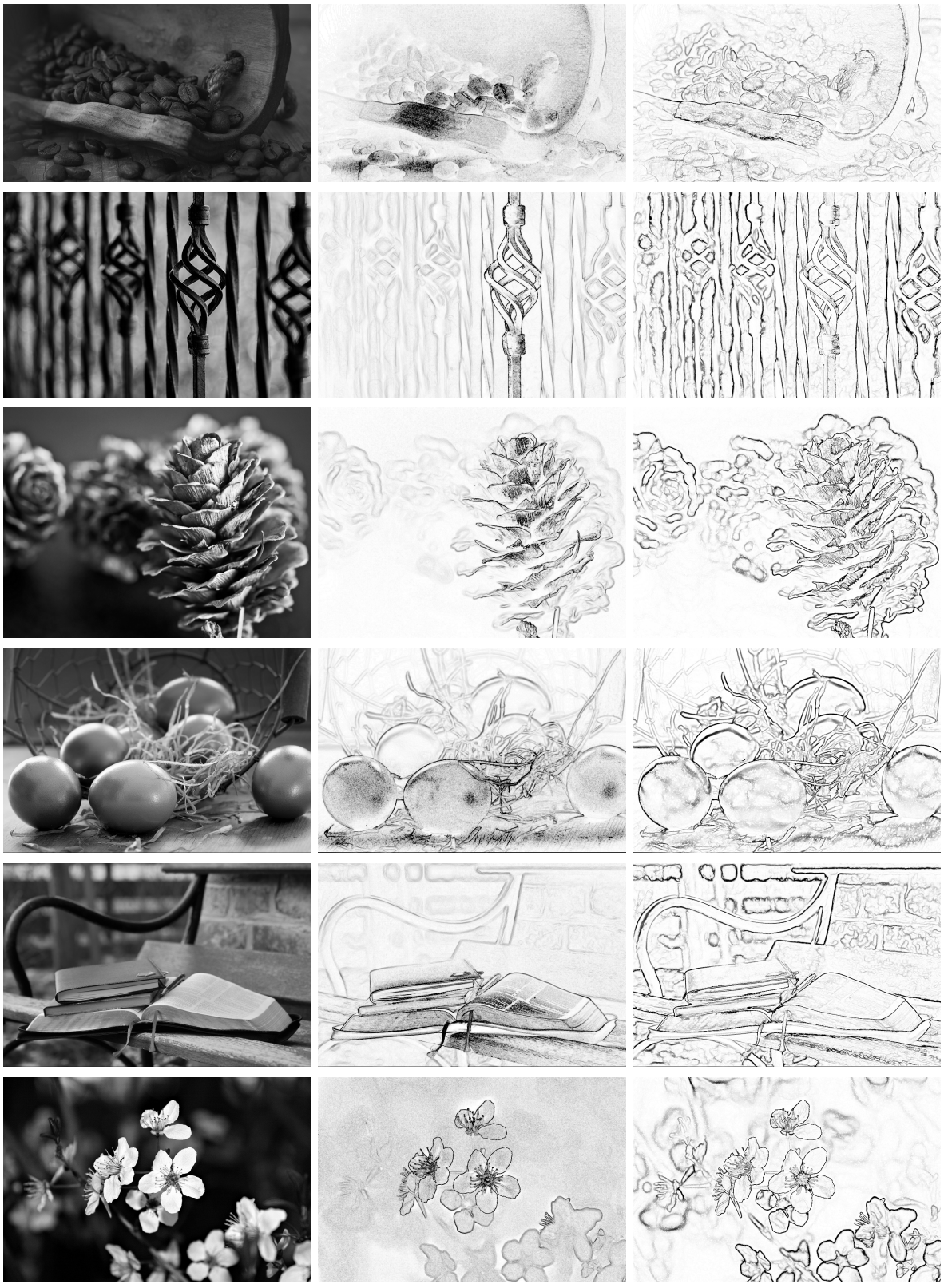
4.1.2 Exploitation of colour information

As mentioned earlier in this text, there are many different ways to compute colour gradients. Adapting morphological operators to operate on colours remains a serious active research topic. The reason behind this is that there is no obvious way to order colours and to make sense of dilations or erosions on colours without a specific application domain. However, when we compute gradients, it is the distance between any two colours in which we are interested. Therefore, the thick gradient of size λ at a given point could easily be replaced by the highest colour distance (for instance, the CIE-LAB ΔE distance) observed between the given point and any of its neighbours at scale λ.

In this section, we focus on *broadcast images*, for which we initially needed the saliency of contours to be enhanced.

Gradient magnitude of perceived brightness The first aspect of the problem was to ensure that the gradient magnitudes obtained in terms of image lightness, truly correspond to that which humans perceive. For example, if we look at the penguin's stomach in image 4.1($\boldsymbol{a}$), we perceive it as being white, while the penguin's leg is black. Therefore, we expect a gradient of very high magnitude at the border separating these two regions. Unfortunately, if we just compute the gradient based on the numerical values stored in each image pixel, we get a low image gradient magnitude. This can be justified by the fact that human perception of the brightness of a greyscale patch follows Weber-Fechner's law. More precisely, the perceived brightness of a greyscale patch is related to its actual luminance by a logarithmic transformation function, except when dealing with luminance values which are critical in imaging. Therefore, when computing the gradient associated with the perceived brightness, we take into account this logarithmic transformation.

Exploitation of the saturation channel A previous study, which is detailed in the article of [Demarty and Beucher, 1998], revealed that a distinction between very desaturated image pixels and those which are highly colourful, usually provides pertinent segmentation boundaries in broadcast images. These boundaries can be perceived as sharp transitions occurring in the images which result from a colour transformation. We refer to this as the enhanced HLS transformation: every image pixel is classified into two groups according to a chosen saturation threshold.



(*a*) (*b*) (*c*)

FIGURE 4.6: Comparison between (*b*) morphological gradients and (*c*) the regularised gradients obtained using algorithm 4.2 on (*a*) defocused greyscale images. The average image dimension is 1920×1276 pixels. The minimum and maximum scales for the regularised gradients are $\lambda_s = 1$ pixel and $\lambda_e = 16$ pixels. Image sources: `https://pixabay.com`.

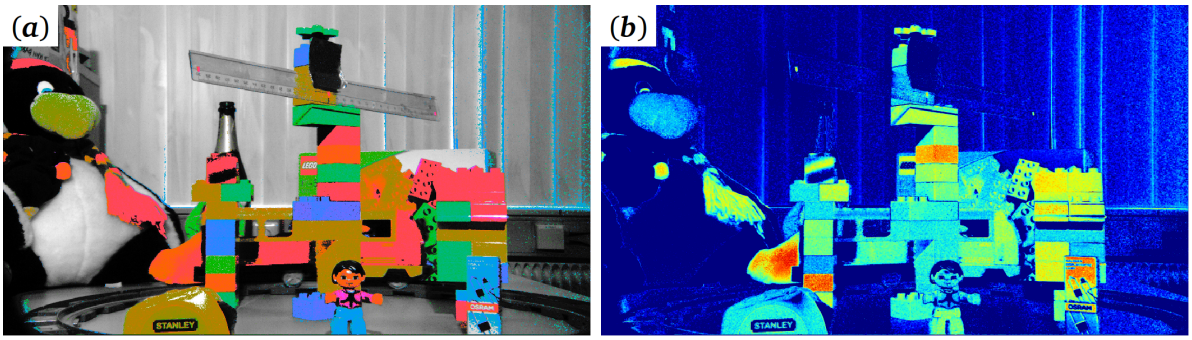


FIGURE 4.7: Saturation helps with discriminating objects in broadcast images. (*a*) The output of the enhanced HLS transformation with a threshold at 20% of the full saturation range reveals interesting frontiers between the saturated and desaturated tones. However, too many artefacts prevents its use for the gradient computation. (*b*) The processed saturation channel to be used within the colour gradient computation.

The pixels with a saturation value below that threshold become completely desaturated in the transformed image. Those with a saturation above the threshold preserve their hue, but are set maximum saturation level and middle lightness level. An example of HLS transformed image is provided in figure 4.7(*a*). One drawback with this transformation is the occasional binarisation effect resulting from the pixel-based classification. Therefore, we propose to apply a non-linear transformation on the saturation function, such that the continuity of the saturation function is preserved, and the rate at which the saturation function increases is more significant when approaching the chosen saturation threshold. To proceed, we use a threshold τ set to 20% of the maximum saturation value so as to transform any saturation value $s \in [0,1]$ into $s' \in [0,1]$ as follows: if $s \geqslant \tau$, then $s' = s$, otherwise $s' = \max(0, (s + (1-\tau))^2 - (1-\tau))$. This yields the saturation channel visualised in figure 4.7(*b*) which can be easily related to the enhanced HLS transformed image.

Combining perceived brightness and saturation information After the logarithmic transformation, the regularised gradients are computed independently for the red and green channels and the altered saturation channels. Finally, the supremum of all these three gradients constitutes the final colour gradient. Figure 4.8 shows the result of this procedure for different types of gradients.

This algorithm concludes our study on the multi-scale enhancements of contour saliency. The next sections concentrate on how we can employ such gradient functions to compute segmentations appropriate for stereo image analysis.

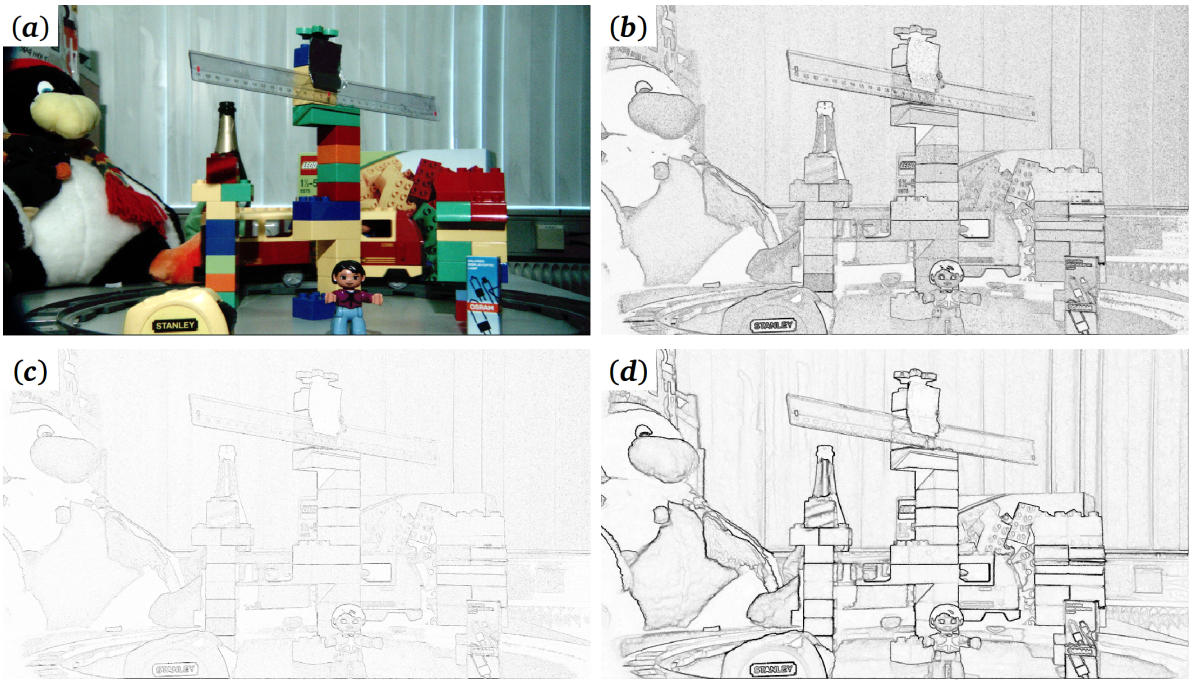


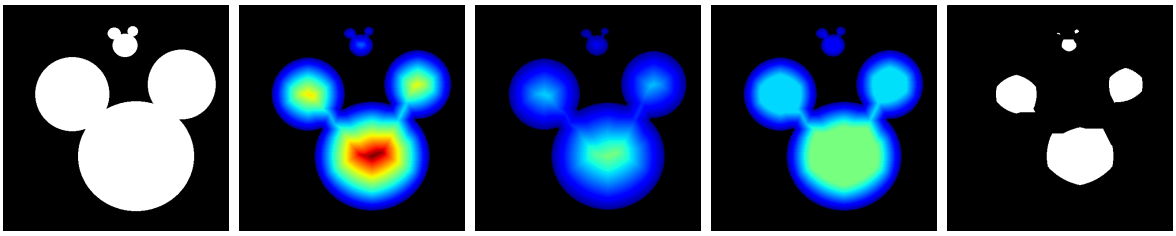
FIGURE 4.8: Comparison between the maxima of the (*b*) morphological gradients, (*c*) regularised gradients and (*d*) enhanced regularised gradients, generated for the red, green and altered saturation channels of (*a*) the input image. For the regularised gradients, the minimum and maximum scales are $\lambda_s = 2$ pixels and $\lambda_e = 10$ pixels.

4.2 Methods of markers generation

The concept of markers controlling segmentations based on the watershed was introduced in section 3.3. Now we are interested in the extraction of markers facilitating the partitioning of an image into a set of homogeneous regions, which potentially constitute good aggregation supports for stereo image analysis. Three methods are investigated: the first consists of converting h-minima of the input gradient to markers, as a result of an adaptive erosion process, whilst the second and third alternatives amount to processing the gradient so that its minima directly constitute the final markers.

4.2.1 Adaptive erosion on gradient's h-minima

In section 3.2, we saw that the computation of the h-minima of a topographical surface, amounts to flooding that topographical surface, such that the depth of the resulting lakes equals h units of elevation maximum. If we interpret the topographical surface as an image gradient, then the h-minima will constitute binary markers overlapping the image areas which have a low contrast. It is therefore tempting to use h-minima as markers for the segmentation of highly salient objects, with respect to the watershed transformation. But in practice, the gradient along the contours of



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

FIGURE 4.9: Adaptive erosion. (a) Input binary mask and (b) its distance function $\mathcal{D}$. (c) Scaled version of the distance function, using $\alpha = \frac{1}{2}$. (d) The geodesic reconstruction of $\mathcal{D}$ by $\alpha\mathcal{D}$. (e) The adaptive erosion of (a) consists of the pixels having different distance values in (b) and (d). The advantage of the adaptive erosion over the regular erosion resides in its ability to mark all the original connected components.

such salient regions, may be low at some locations, due to a locally poor contrast. The proposed algorithm is used to handle what we call *gradient leakages* when computing markers based on the h-minima of the image gradient.

This problem shares some similarities with the segmentation of coffee beans discussed in [Vincent and Dougherty, 1994], that is, the segmentation enabling the separation of coffee beans based on the ultimate erosion of a binary set. Leakages are indeed characterised by a narrowing of the h-minima. Applying an erosion to the original shape of the h-minima therefore splits the markers where leakages occur, but a significantly strong erosion would also destroy pertinent markers across thin regions, hence the need for an adaptive erosion. Suppose that $\mathcal{D}$ is the distance function associated with the binary image $\mathbf{M}_h(\mathbf{I})$ containing the h-minima of the gradient image $\mathbf{I}$, as in equation 3.13. Let $\tilde{\mathbf{M}}_h(\mathbf{I}, \alpha)$ be the marker set obtained after an adaptive erosion on $\mathbf{M}_h(\mathbf{I})$ of strength $0 \leqslant \alpha < 1$, where α is a real number. The set of points activated in $\tilde{\mathbf{M}}_h(\mathbf{I}, \alpha)$ is expressed by equation 4.4.

$$S(\tilde{\mathbf{M}}_h(\mathbf{I}, \alpha)) = \{(x, y) \mid (\mathcal{D} - \mathrm{R}_{\mathcal{D}}(\alpha\mathcal{D}))[x, y] > 0\} \tag{4.4}$$

Figure 4.9 illustrates how the reconstruction of the original distance function by a scaled version of the latter yields the output markers. α controls the adaptive erosion's strength. When $\alpha = 0$, the original h-minima remain unaltered. As $\alpha \to 1$, the markers resulting from the adaptive erosion process tend towards the ultimate erosion of $\mathbf{M}_h(\mathbf{I})$. Furthermore, since $\tilde{\mathbf{M}}_h(\mathbf{I}, \alpha)$ is the residue of a geodesic reconstruction of $\mathbf{M}_h(\mathbf{I})$, the inclusion property $S(\tilde{\mathbf{M}}_h(\mathbf{I}, \alpha)) \subseteq S(\mathbf{M}_h(\mathbf{I}))$ holds, and more generally $S(\tilde{\mathbf{M}}_h(\mathbf{I}, \alpha')) \subseteq S(\tilde{\mathbf{M}}_h(\mathbf{I}, \alpha))$ for $\alpha' \geqslant \alpha$. However, the hierarchical relationship obtained for the h-minima in function of the elevation h is no longer verified: indeed, $h' < h$ does *not* necessarily imply that $S(\tilde{\mathbf{M}}_{h'}(\mathbf{I}, \alpha)) \subseteq S(\tilde{\mathbf{M}}_h(\mathbf{I}, \alpha))$.

There is nonetheless a method which permits the generation of hierarchical segmentations based on adaptively eroded h-minima. Suppose $\mathbf{M}_W$ is the binary image containing the wa-

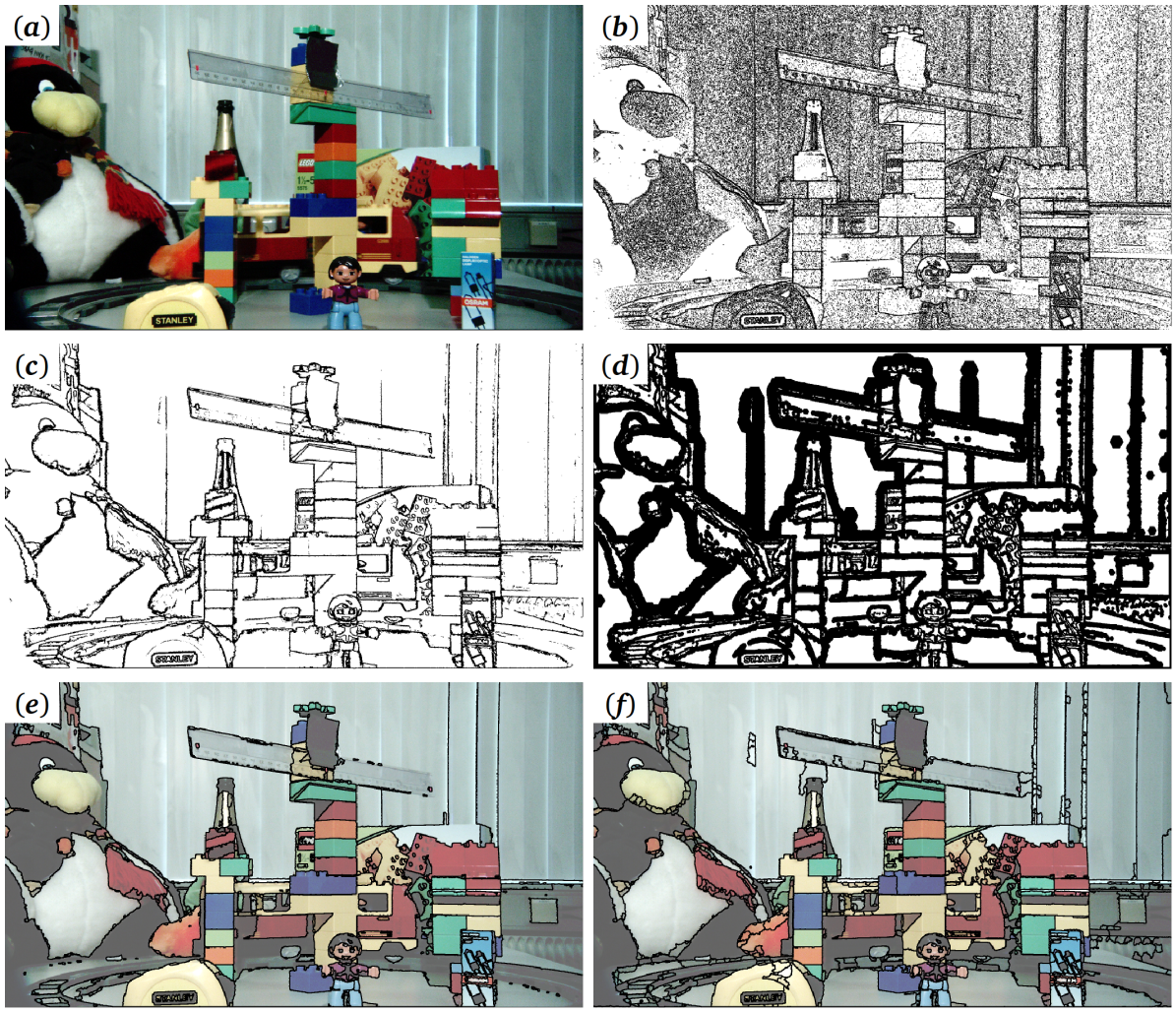


FIGURE 4.10: Segmentation examples using the adaptive erosion on h-minima with an elevation of $\mathrm{h} = 20$. (a) Input image, (b) The h-minima of the basic colour gradient are practically useless, whereas (c) the h-minima of the enhanced regularised gradient seem to highlight the different homogeneous regions quite well. Unfortunately, leakages in the gradient lead to the early fusion of catchment basins, hence (e) the resulting segmentation missing many important object boundaries (ruler, railway lines, etc.). It is possible to apply (d) an adaptive erosion on the h-minima, in order to obtain (f) a slightly over-segmented but more useful partition. The scaling factor employed in this example was set to $\alpha = 0.25$.

tershed obtained using $\tilde{\mathbf{M}}_h(\mathbf{I}, \alpha)$ as the segmentation markers. We would like to generate a *finer* segmentation containing the watershed held by $\mathbf{M}_\mathcal{W}$. To proceed, let us define a new topographical surface $\mathbf{I}'$ such that:

$$\mathbf{I}'[x,y] = \begin{cases} \mathbf{I}[x,y] & \text{if } (x,y) \notin S(\mathbf{M}_\mathcal{W}) \\ +\infty & \text{otherwise} \end{cases}$$

Notice that the points of $S(\mathbf{M}_\mathcal{W})$ belong to the catchment basins' frontiers of the topographical surface $\mathbf{I}'$. Besides, since they are of an infinite altitude, we have the guarantee that each catchment basin of $\mathbf{M}_\mathcal{W}$ is marked by at least one h-minimum of $\mathbf{I}'$ and that none of these h-minima crosses the frontier of a catchment basin in $\mathbf{M}_\mathcal{W}$. Therefore, if $\mathbf{M}'_\mathcal{W}$ contains the segmentation controlled by the topographical surface $\mathbf{I}'$ and the markers described by the binary image $\tilde{\mathbf{M}}_{h'}(\mathbf{I}', \alpha')$ for $h' < h$ and $1 > \alpha' \geqslant \alpha$, then $S(\mathbf{M}_\mathcal{W}) \subseteq S(\mathbf{M}'_\mathcal{W})$ as desired. The usage of hierarchical segmentations within stereo analysis will become clearer when dealing with the refinement of disparity measurements.

To summarise, the adaptive erosion of the gradient's h-minima produces pertinent markers of contrast. However, they may not solve our segmentation problem completely: for example, it could be that the gradient contains *parasites*, i.e. very small, yet contrasted catchment basins, which would be retained in the final segmentations. Some morphological filtering, such as the supremum of directional openings on the marker set, could prove useful to avoid the segmentation of these catchment basins. The following section focuses on reconstruction and chaining mechanisms enabling the generation of markers based on criteria other than the contrast alone.

4.2.2 Criteria-based reconstructions of the gradient and minima

In this second alternative, the segmentation markers correspond to the minima of a topographical surface resulting from a dual reconstruction of the original gradient. To be more specific, the dual reconstructions discussed here fall into the category of synchronous flooding, which has been thoroughly discussed in [Zanoguera Tous, 2001] and [Gomila, 2001]. The most straightforward type of synchronous flooding is that based on the depth of the catchment basins, i.e. a flooding from which the h-minima of the gradient are typically recovered. The method we now introduce as algorithm 4.3, is designed to simulate the depth-based flooding of the original topographical surface represented by the image gradient $\mathbf{I}$. As the flooding continues, we track the evolution of the attributes associated with the constructed lakes and catchment basins. For any given lake, the flooding may be permanently interrupted once the minimum depth for which the associated attributes satisfy the chosen segmentation *criterion*, has been attained.

Algorithm 4.3 Criterion-based flooding of a topographical surface

```
1: function BUILDONCRITERION(I, criterionFx)
2:     h ← 0
3:     S_h ← I                                   ▷ Current state of the flooding
4:     while h ⩽ h_max do
5:         M_h ← criteronFx(I, S_h)              ▷ Selection mask of "invalid" catchment basins
6:         if ∃(x, y) | M_h[x, y] ≠ 0 then
7:             S_h ← Fl(S_h, S_h + M_h)          ▷ Flooding on selected catchment basins
8:         h ← h + 1
9:     return S_h
```

The segmentation criterion refers to a threshold, for instance an objective with respect to the depth, the area, or the volume of the lakes produced by the flooding. Suppose we are given the original image gradient $\mathbf{I}$ as well as a flooding of that image, say $\mathbf{S}_h$. The zones of $\mathbf{I}$ being flooded are given by the set of points $S = \{(x, y) \mid (\mathbf{S}_h - \mathbf{I})[x, y] > 0\}$. Furthermore, let $\mathcal{L}_h$ be the image mapping every pixel (x, y) to the label $\mathcal{L}_h[x, y]$ of the corresponding catchment basin of $\mathbf{S}_h$. We denote by $\mathbf{R}_i$, the region describing a particular catchment basin of $\mathcal{L}_h$ as $S(\mathbf{R}_i) = \{(x, y) \mid \mathcal{L}_h[x, y] = i\}$. Then, the following attributes can be computed for each catchment basin associated with $\mathbf{R}_i$:

- the lake depth d_i, given by the equation $d_i = \max_{(x,y)\in S_i} (\mathbf{S}_h - \mathbf{I})[x, y]$
- the lake volume v_i, given by the equation $v_i = \sum_{(x,y)\in S_i} (\mathbf{S}_h - \mathbf{I})[x, y]$
- the area a_i of the catchment basin, related to the cardinal of the set $\mathbf{R}_i$ by $a_i = \sum_{(x,y)\in S_i} 1$

Based on the combination of such attributes, one may define more complex criteria. For example, the volume attribute yields perceptually appealing segmentations. However, very large homogeneous regions have a tendency to become over-segmented whilst particularly thin structures never appear in the final segmentation despite their contrast. To overcome that limitation, it is necessary to impose a volume limit v, in addition to ensuring that the non-absorbed lakes have a minimum depth of d_{min}, while not exceeding the maximum depth of d_{max} elevation units. At iteration h, the binary mask highlighting the pixels belonging to the lakes satisfying such conditions, is given by relation 4.5.

$$\mathbf{M}_h[x, y] = \bigcup_i (\mathcal{L}_h[x, y] = i) \wedge ((d_i \leqslant d_{min}) \vee ((v_i \leqslant v) \wedge (d_i \leqslant d_{max}))) \tag{4.5}$$

The use of such a binary image, for a given elevation value h, is key to the success of the flooding process represented by algorithm 4.3. At each iteration step h, the catchment basins which do not *yet* satisfy the chosen criterion, are selected by function `criterionFx` and represented by means of a binary image; for instance $\mathbf{M}_h$, which constitutes the output of the function. In general, the criterion is chosen so that $S(\mathbf{M}_h) \supseteq S(\mathbf{M}_{h+1})$, i.e. it is not possible for the flooding state of a catchment basin to first verify and then contradict the criterion. Finally, as desired, the depth of

the lakes increases only for those appearing in this validity mask. Figures 4.11 and 4.12 illustrate area- and volume-based flooding using constraints derived from the lake depths.

Synchronous flooding methods have enjoyed a wide range of applications in image compression. Since they induce hierarchies of segmentations [Meyer, 2001], it is indeed possible to search a hierarchy for a partition composed of a fixed number of regions. The pertinence of these regions, of course, will depend on the segmentation criterion. However, in an application like ours, where the delineation of every objects is important, limiting the number of regions would not be realistic. It is possible to manually set the contrast, area or volume criteria for a single-shot video sequence, but one should not expect such fixed parameters to remain optimal when used across a database composed of diverse images. It would be useful to add a criterion based on the consistency between the mosaic image resulting from the current segmentation and the original image, as in [Vilaplana et al., 2008]. However the problem would be different from ours, since it would be necessary to stop the flooding *before* the criterion was breached. There is also a growing interest in employing energy functions to determine optimal cuts across trees encoding partition hierarchies [Kiran and Serra, 2014], but here too the success of the method depends on how the hierarchies are constructed and on the pertinence of the energy function symbolising, in that case, the segmentation criterion.

The last method of generating markers presented in this section resorts to a viscous transformation applied to the input gradient. The idea is to devise an algorithm, which can be applied to any kind of image, and which is able to produce more relevant over-segmentations than the watershed resulting from the direct minima of the gradient. In section 4.3, we show how the regions of such over-segmentations can be merged in order to provide consistent correlation supports.

4.2.3 Viscous transformation of the gradient and minima

Originally, the purpose of the viscous watershed transformation was to compute the watershed of an arbitrary topographical surface, using a viscous substance in place of water within the uniform flooding procedure. In other words, the traversal of the topographical surface by the viscous liquid at a given altitude is impeded in sinuous and narrow passages, thus making the watershed less sensitive to gradient leakages. In their work, [Vachier and Meyer, 2005] show that simulating a viscous flooding on the original topographical surface $\mathbf{I}$ amounts to performing a standard flooding on an altered version of the topographical surface, say $\tilde{\mathbf{I}}$. In the case of oil-based flooding, the construction of $\tilde{\mathbf{I}}$ consists of applying closings of different sizes to each level set $\mathbf{I}_0, \ldots, \mathbf{I}_n$ of $\mathbf{I}$, while decreasing the closing strength λ_i as the altitude increases. Consequently, the resulting contours are quite regular if they happen to cross homogeneous areas while sharp gradient areas remain delineated with high fidelity in terms of boundary adherence. Mathematically speaking, $\mathbf{I}$

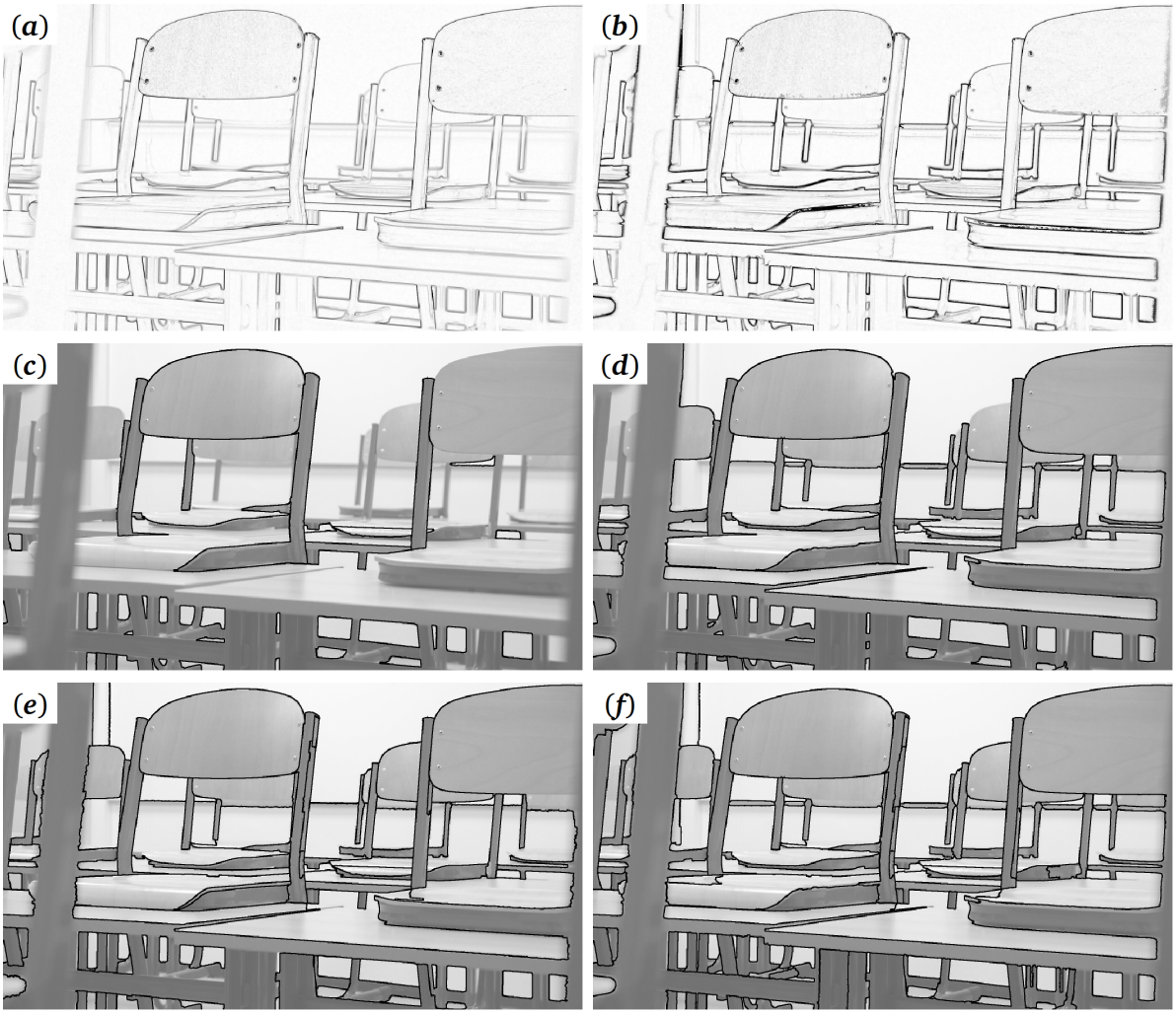


FIGURE 4.11: Impact of regularised gradient and chaining between contrast and area attributes on image segmentation. (*a*) Morphological gradient, (*c*) the resulting segmentation from the h-minima of the gradient, for $\mathrm{h} = 25$, (*e*) the resulting segmentation from a synchronous flooding based on a fixed area criterion, imposing a lake of minimum depth $\mathrm{h} = 5$ and maximum depth $\mathrm{h} = 25$. (*b*) Enhanced regularised gradient. (*d*) and (*f*) correspond to the resulting segmentations using the same parameters as in (*c*) and (*e*) respectively. Image source: `https://pixabay.com`

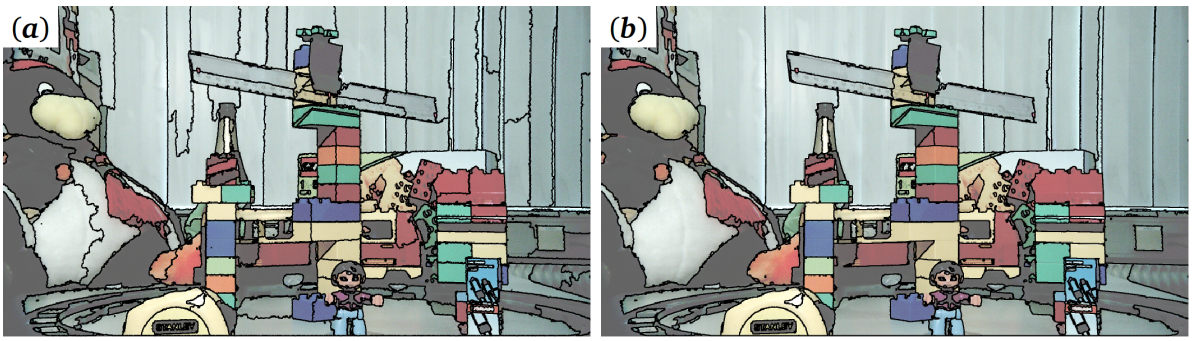


FIGURE 4.12: Segmentations resulting from the flooding of the gradient shown in figure 4.8(*d*). Both are controlled by the criterion of equation 4.5 with (*a*) $\mathrm{d}_{\min} = 0$, $\mathrm{d}_{\max} = 25$ and (*b*) $\mathrm{d}_{\min} = 10$, $\mathrm{d}_{\max} = 25$.

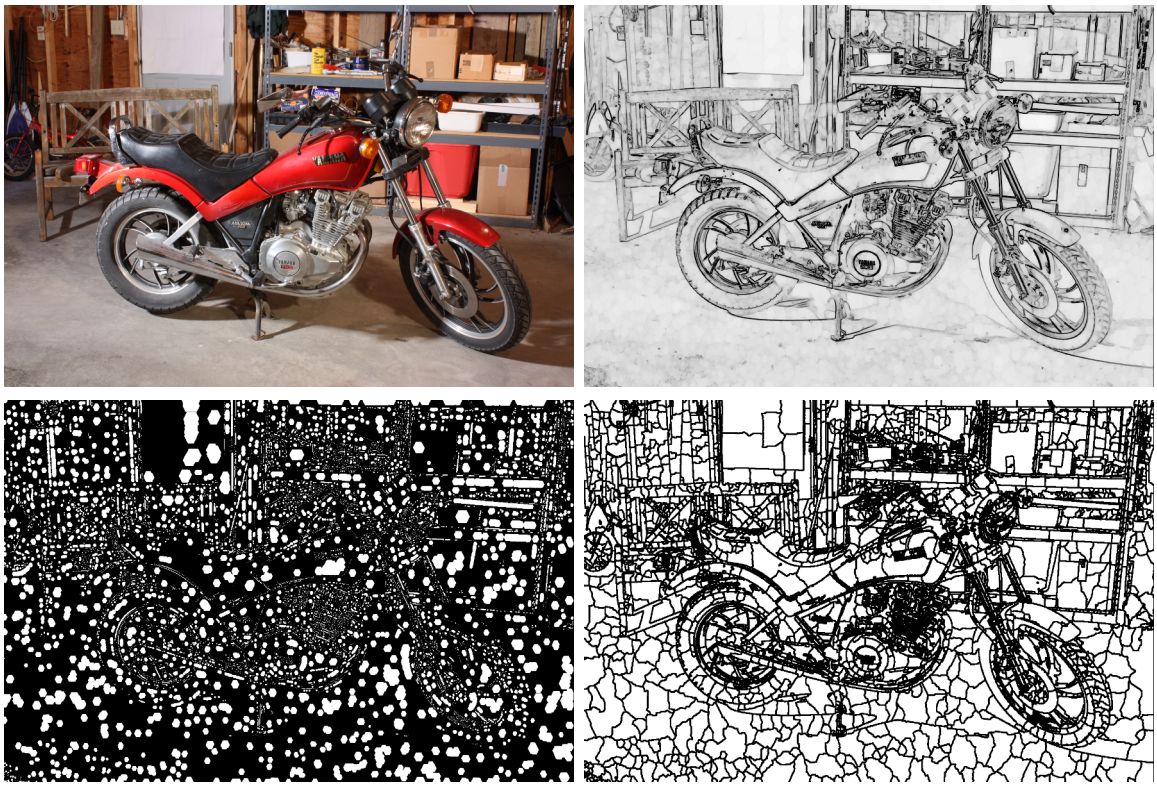


FIGURE 4.13: Viscous transformation of a gradient, based on oil flooding. Top row: input image and viscous gradient. Bottom row: the minima of the viscous gradient and the resulting segmentation of the gradient's catchment basins. Image source: `Motorcycle` from Middlebury 2014 dataset.

and $\tilde{\mathbf{I}}$ are related by equation 4.6.

$$\tilde{\mathbf{I}} = \sup_{i} \{ i \times \varphi_{\lambda_i \mathbf{H}} (\mathbf{I}_i) \} \tag{4.6}$$

In our experiments, the choice of λ_i has been determined empirically: we express the latter as an exponential decay of the form $\lambda_i = \lfloor 25 \cdot \exp(-0.1 \cdot i) + \frac{1}{2} \rfloor$, where i denotes the greyscale level in the range of 0 and 255. Therefore, the viscous transformation we perform only alters the gradient levels for $i < 40$, with a closing strength decreasing from $\lambda_1 = 23$, to $\lambda_{10} = 9$, to $\lambda_{20} = 3$, etc.

Figure 4.13 shows an example of such a viscous transformation applied to an image gradient. We can observe that the markers yield a perceptually consistent over-segmentation. Nonetheless, some segments are extremely small, especially across textured areas, making them inadequate if used as correlation supports. The last section of this chapter therefore presents a segmentation algorithm which resorts to both the viscous transformation and the contrast and area criteria to obtain useful over-segmentations.

4.3 Over-segmentation for correlation analysis

In the over-segmentations resulting from the direct viscous watershed transformation, we observe that homogeneous regions are fragmented into many cells, without any clear meaning with respect to the perceived regional boundaries. Such a partitioning is not desirable, if our goal is to enforce consistency of stereo matches across homogeneous areas. Fortunately, the gradient values associated with the non-relevant boundaries are very low and therefore can easily be removed. A second source of severe over-segmentation arises from the textured areas, where cells are particularly small. Although stereo matches tend to be more achievable across such areas, parasite cells do not offer sufficiently discriminant correlation supports. Therefore, a minimum cell area must be enforced to facilitate the establishment of future matches. Obviously, we are again interested in a criterion-based segmentation. However, unlike the method proposed in section 4.2.2, the method proposed in this section:

1. employs the viscous transformed gradient, in order to pre-empt the problem of leaking passages.
2. performs the segmentation without having to simulate the complete synchronous flooding. This latter would indeed be time consuming, since all depths would need to be tested to ensure that the minimum area criterion remained predominant.
3. automatically tunes the contrast and area thresholds based on the image contents.

The algorithm

The proposed algorithm is illustrated by the flowchart displayed in figure 4.14. As input, the segmentation process requires $\tilde{\mathbf{I}}$, the image gradient obtained after the viscous transformation described in section 4.2.3. Then two partitions $\mathcal{L}_c$ and $\mathcal{L}_f$ are generated. The first is designed to contain regions which split only at very sharp edges, while satisfying the minimum area criterion. Alternatively, the second is composed of segments with less accentuated boundaries, but with more significant areas.

$\mathcal{L}_c$ and $\mathcal{L}_f$ are computed by means of a watershed transformation of $\tilde{\mathbf{I}}$. In both cases, the markers driving this transformation are obtained by a simple threshold on $\tilde{\mathbf{I}}$. The threshold parameters t_c and t_f with $t_c > t_f$ symbolise therefore the minimum gradient values expected along the boundaries of the regions appearing in $\mathcal{L}_c$ and $\mathcal{L}_f$ respectively. Next, the partition-based area openings, controlled by parameters σ_c and σ_f with $\sigma_c < \sigma_f$, prune the cells in $\mathcal{L}_c$ and $\mathcal{L}_f$ which have not attained an area of σ_c and σ_f pixels respectively. With respect to figure 4.14, we have just reached breakpoints ($\boldsymbol{a}$) and ($\boldsymbol{c}$). As can be observed, the generation of $\mathcal{L}_c$ involves more work than that of $\mathcal{L}_f$. Indeed, both partitions contain holes, i.e. pixels with a label value set to zero. So, if a textured image zone is composed of a multitude of small regions which do

not satisfy the minimum area criterion depending on σ_c, then it will be necessary for the entire textured zone to be replaced by a large hole in the partition obtained at breakpoint (***a***). The partition map obtained at breakpoint (***b***) represents all the holes resulting from the small area opening on a labelled partition map. The holes with an area higher than σ_c pixels are recovered and appended to the partition obtained at breakpoint (***a***), which constitutes the final construction step resulting in partition $\mathcal{L}_c$.

The final part of the segmentation procedure consists of building a new map of markers from partitions $\mathcal{L}_c$ and $\mathcal{L}_f$. In fact, thin and elongated regions as well as textured regions should have disappeared from partition $\mathcal{L}_f$ because of the high area criterion. $\mathcal{L}_c$ is the only partition able to represent these regions, provided that their boundaries are sufficiently accentuated in the gradient $\tilde{\mathbf{I}}$. We are therefore interested in the residue of reconstruction of the cells [Beucher, 2013a] of $\mathcal{L}_c$ by the cells of $\mathcal{L}_f$, since this residue contains all the cells in $\mathcal{L}_c$ which were replaced by holes in $\mathcal{L}_f$. The structures appearing in this residue, at breakpoint (***d***), are ultimately added to $\mathcal{L}_f$ in order to constitute the new map of markers, which in conjunction with $\tilde{\mathbf{I}}$, will drive the final watershed segmentation.

Figure 4.15 highlights some breakpoints of the segmentation procedure performed on an example composed of a large number of thin and textured regions, while figure 4.16 provides additional examples performed on images from the Middlebury 2014 database.

Automatic tuning of parameters

The selection of the four area and contrast parameters will depend on the image characteristics. In our experiments, t_c corresponds to the 45^{th} percentile of the intensity values observed in the initial colour gradient excluding 0, and t_f corresponds to the 10^{th} percentile. Additionally, the ratio of the opening parameter with respect to the full image area is fixed. In our experiments, this ratio has been set to $5 \cdot 10^{-5}$ for σ_c and to ten times more for σ_f.

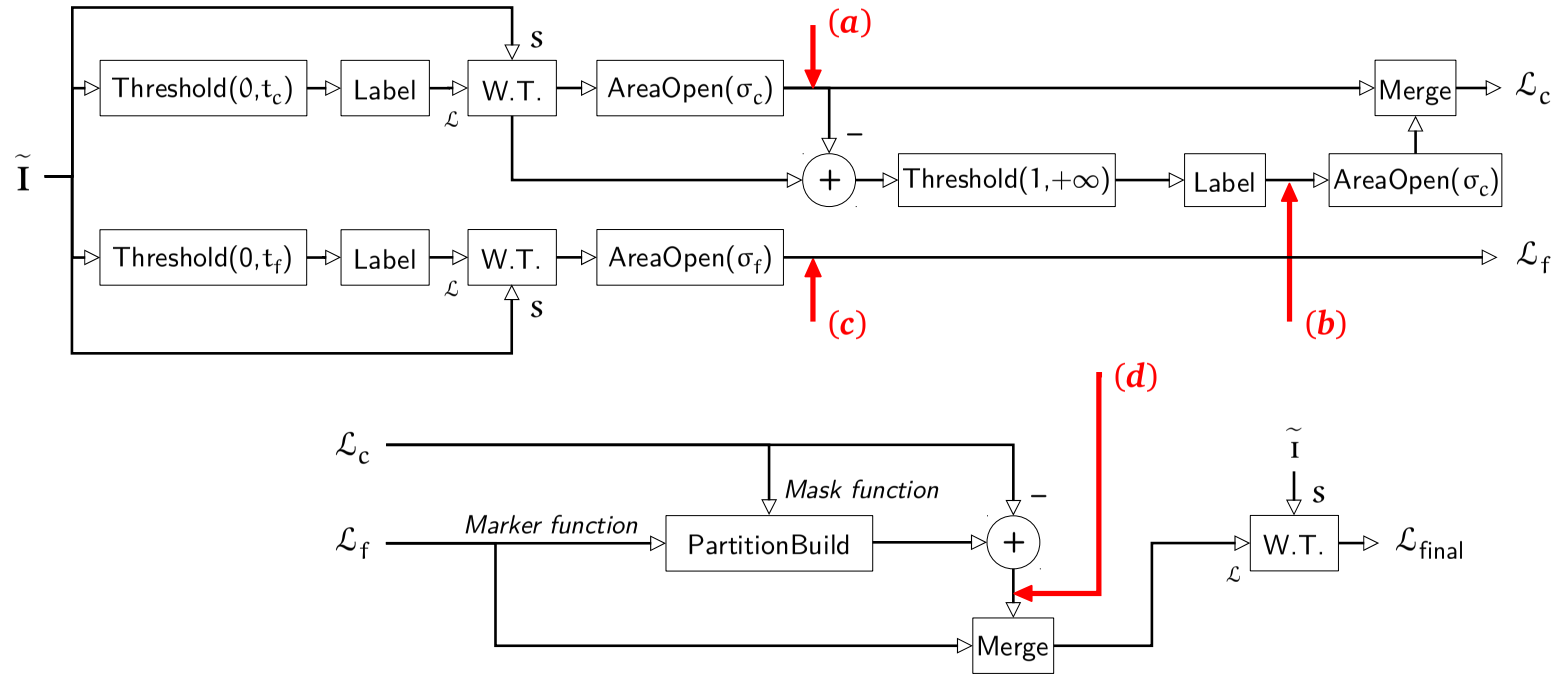


FIGURE 4.14: Flowchart of the image over-segmentation procedure.

- AreaOpen(σ) is an area opening operator pruning any cell of the partition having an area inferior or equal to σ pixels.
- Label is a labelling operator which, given a binary image as input, assigns a unique label to each of its connected components. Pixels set to 0 in the binary image are set to 0 in the labelled image.
- Merge is responsible for merging two images of markers while ensuring that each marker receives a unique label in the output image.
- PartitionBuild takes two images of lakes $\mathcal{L}_{\text{mask}}$ and $\mathcal{L}_{\text{marker}}$ as input. The output $\mathcal{L}_{\text{out}}$ is an image of lakes of identical dimensions, defined by the following relation:

$$\mathcal{L}_{\text{out}}[\mathbf{p}] = \begin{cases} \mathcal{L}_{\text{mask}}[\mathbf{p}] & \text{if } \exists\, \mathbf{p}' \mid \mathcal{L}_{\text{mask}}[\mathbf{p}'] = \mathcal{L}_{\text{mask}}[\mathbf{p}] \wedge \mathcal{L}_{\text{marker}}[\mathbf{p}'] > 0 \\ 0 & \text{otherwise} \end{cases}$$

- Threshold(t_0, t_1) denotes the standard threshold operator mapping all pixels of the input image in the range $[t_0, t_1]$ to 1, the others to 0.
- W.T. is the watershed transformation described in section 3.3.

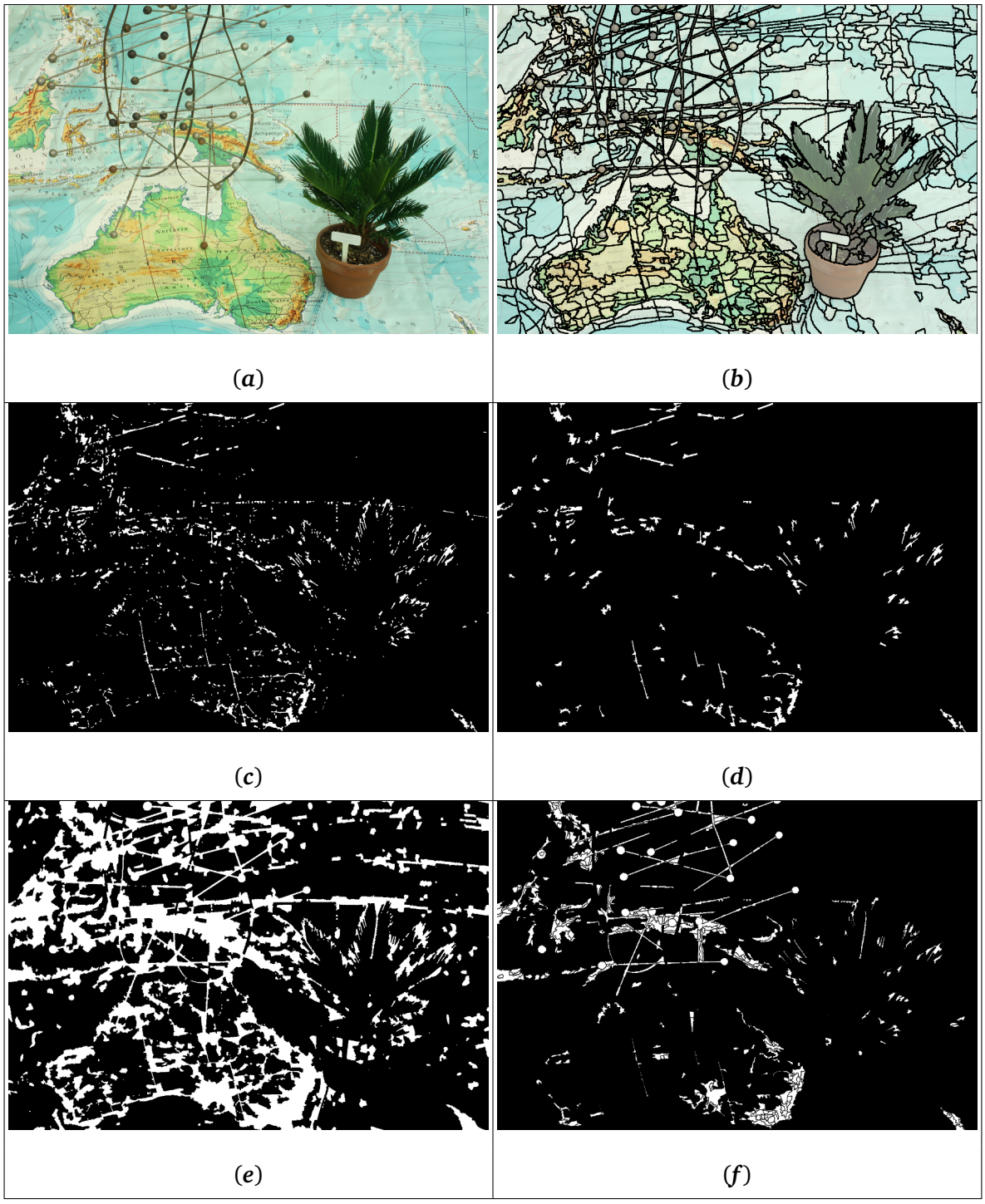


FIGURE 4.15: The over-segmentation of `AustraliaP`. (a) Input image and (b) its associated over-segmentation. (c) Binary mask highlighting the structures removed by the area opening at breakpoint (a) (cf. figure 4.14). Some of these structures are either parasites, or they constitute parts of thin objects, or parts of textured regions. (d) The holes marking new regions in the partition image $\mathbf{M}_c$. (e) The holes appearing in partition image $\mathbf{M}_f$, (f) Binary mask highlighting the markers in $\mathbf{M}_c$ which are merged with those of $\mathbf{M}_f$ at breakpoint (d). This last step of the over-segmentation algorithm is essential to ensure that the thin and salient regions appear in the final segmentation.

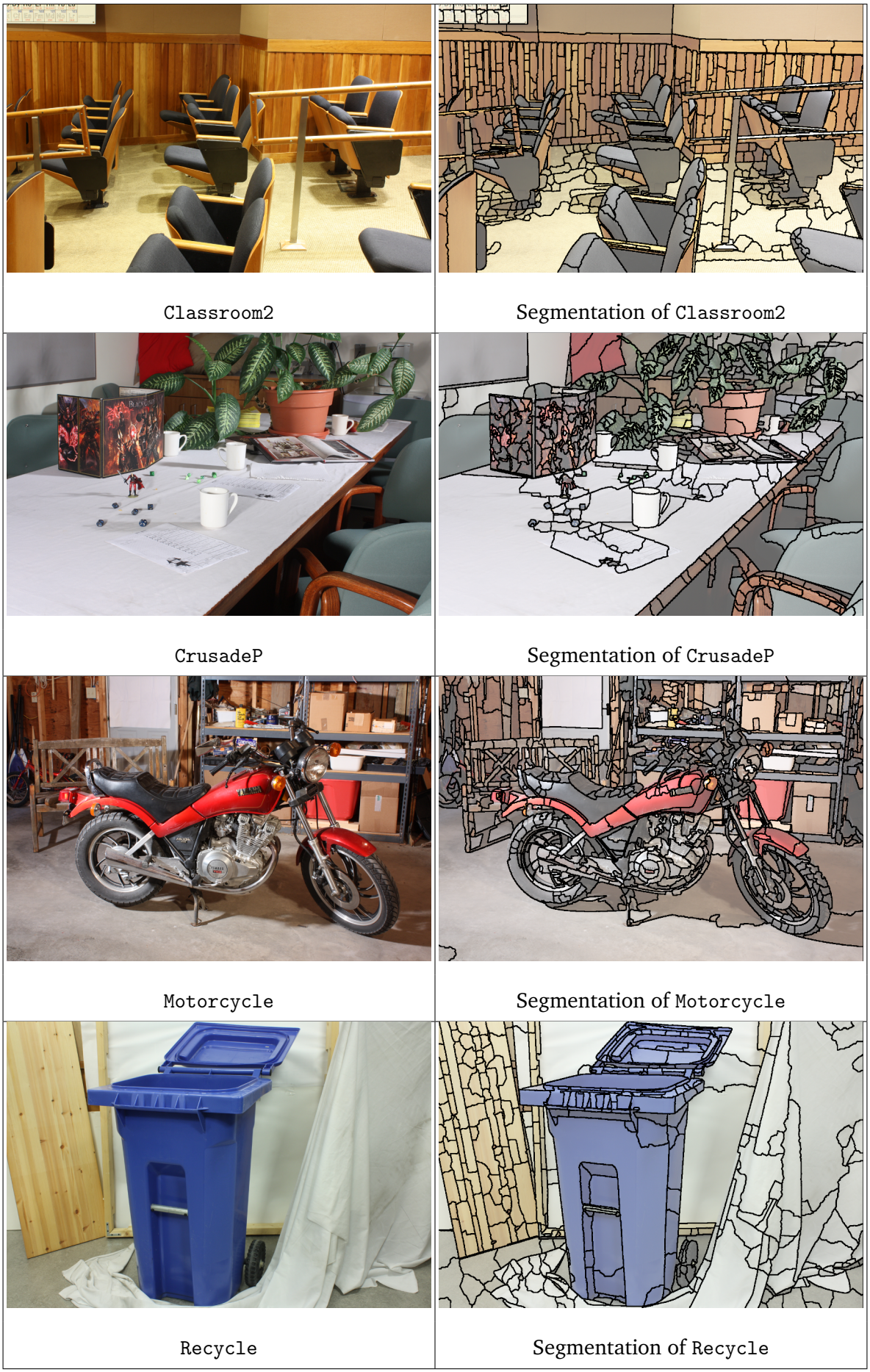


FIGURE 4.16: Some examples of over-segmentations on Middlebury 2014 dataset.

Summary

In order to harness the watershed transformation so that it produces pertinent image segmentations, it is necessary to use care when choosing the topographical surface that will be flooded, and the controlling markers.

Regarding the topographical surface, we developed an enhanced version of the regularised gradient, which preserves its multi-scale properties but which also ameliorates gradient values near region boundaries so that they remain consistent with respect to human perception. This type of gradient proves particularly useful when applied to natural scene images where there is partial blurring due to defocus aberration. Its attractive filtering properties mean it is also suitable for non-defocused images.

Next we addressed the problem of automatic markers extraction from the topographical surface of interest. We proposed different approaches based on depth-based synchronous flooding, for which the flooded parts of the image belong to the connected components describing the markers. We saw that it is possible to rely solely on the contrast, but that the markers usually require some post-processing, similar to adaptive erosion, in order to avoid leaking gradient passages, which could prematurely merge regions. Combining contrast with other criteria, such as area, usually leads to more useful and meaningful object-like segmentations.

The concluding part of this chapter focused on controlled image over-segmentations. We produced partitions offering a trade-off between the contrast and area criteria. In particular, the regions each covering a small area of the image plane can be preserved only if they are salient, while the other regions with less accentuated boundaries must cover larger areas. Moreover, the regions of these partitions are always guaranteed a minimum area. The contrast and area parameters driving these over-segmentations have also been adapted to the dynamic range and size of the image respectively. This results in segmentations which will remain consistent across an entire database.

Both over-segmentation and object-like segmentation will make their appearance in our depth map estimation methods, since their regions constitute reasonably wide and consistent aggregation supports with respect to the image contents.

Résumé du chapitre 5

Au chapitre 2, nous avons établi que régions homogènes et occultations étaient source de nombreux problèmes lors de l'établissement d'appariements entre les deux images stéréoscopiques. Dans ce chapitre, nous proposons deux méthodes alternatives permettant de gérer ces deux difficultés, au moyen des segmentations associées aux deux images stéréoscopiques.

Les régions générées pour chaque image stéréo sont exploitées dans le but de contraindre la façon dont les coûts de superpositions peuvent s'agréger. Plus précisément, nous empêchons que les coûts provenant de superpositions de régions différentes puissent se mélanger. De cette manière, les coûts résultant d'une occultation ne peuvent s'agréger avec ceux résultant d'une superposition effective entre deux régions.

Les résultats produits par les deux algorithmes sont cependant très distincts. La première méthode produit des cartes de disparités régionales. Dans ces dernières, tous les pixels d'une même région reçoivent la même mesure de disparité. La fonction de disparité résultante est donc lisse par morceaux. De plus, les discontinuités de disparités délimitent avec précision les différents objets présents dans la scène, et les pixels étant occultés dans l'une des deux images stéréoscopiques reçoivent des valeurs de disparité plausibles. D'un point de vue perceptuel, cela rend ces cartes de disparités régionales particulièrement attrayantes, en particulier lorsqu'une majorité d'objets se positionne de manière quasi fronto-parallèle à la caméra. En revanche, dans les bases de données modernes, la validité de telles hypothèses géométriques est de plus en plus remise en question.

Dans cette optique, nous avons proposé une deuxième approche qui cherche, au moyen d'un volume de superposition d'images filtré, des mesures de disparité sur une base ponctuelle. Le filtrage du volume dépend de deux paramètres : le premier contraint la portée maximale de l'agrégation, le second contrôle les fluctuations des chemins d'agrégation couvrant plusieurs décalages pour lesquels les images stéréo ont été superposées. Nous observerons que, pour une portée d'agrégation suffisamment large, et en utilisant un filtrage morphologique approprié, il est possible d'obtenir des cartes de disparités sporadiques très précises, malgré l'absence de mesures dans les régions homogènes. Cependant, en considérant différentes portées d'agrégation, nous verrons qu'un mécanisme mesurant les disparités à différentes échelles permet d'aboutir à des

cartes de disparités sporadiques beaucoup plus denses, dont les seules informations manquantes se situent au niveau des pixels occultés dans l'autre image de la paire stéréoscopique. Le lecteur trouvera dans ce chapitre les détails nécessaires à la compréhension des paramètres et à l'implémentation du filtrage des cartes de disparités sporadiques. Enfin, un tableau permettra de comparer de manière efficace les deux méthodes proposées.

Chapter 5

Disparity Measurements based on Regions

In this methodology, the extraction of pertinent disparity measures and the estimation of the whole disparity map are dissociated. This chapter therefore deals with the first aspect, i.e. the measurement of disparities across a pair of stereo images. We propose two alternatives based on the segmentations described in chapter 4.

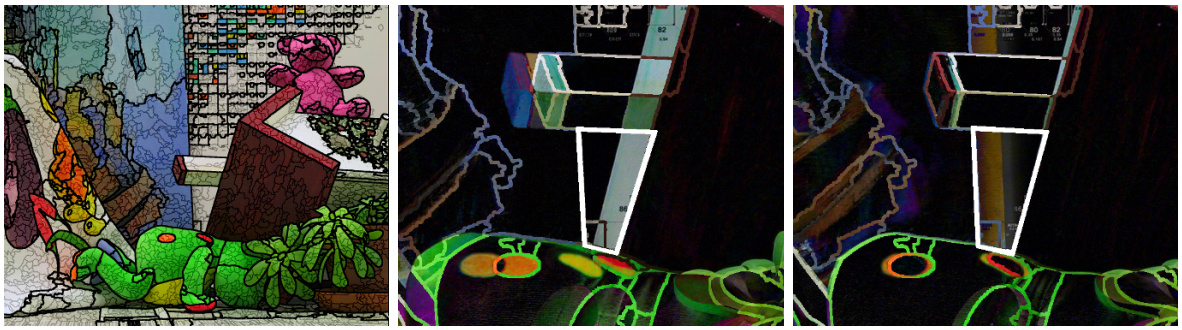
Section 5.1 deals with the first approach resorting to *regional disparities*. These measures enable the generation of consistent disparity maps with respect to depth perception and facilitate refinement operations across segmentation hierarchies. A key advantage of regional disparity maps, if all underlying assumptions of the model are satisfied, is that no estimation procedure is required to deduce a complete disparity map. Section 5.2 presents a second approach which is more robust in cases where both the stereo images were acquired using a wide baseline and the regions composing these images depict very tilted objects in the actual 3D scene. This second alternative however yields sparse disparity maps.

5.1 Regional disparities

A regional disparity is a measure allocated to a region of the left image partition, i.e. the one for which we aim to compute the associated disparity map. It indicates the intensity of the horizontal shift which must be applied to the right image towards the right-hand side, so that the left and shifted right images are in "best" superimposition within the region under consideration.

The computation of regional disparities is initially performed on coarse partitions, computed for example using either of the algorithms introduced in sections 4.2.1 and 4.2.2. Using the notation of table 2.1, we assume that $\mathbf{R}_i$ represents the i-th region of the left image partition, whilst $\mathbf{R}_j'^{(d)}$ corresponds to the j-th region of the right image partition, shifted horizontally towards the right by d pixels. Furthermore, we define $\mathfrak{D}$ as the regional disparity map associated with the left view $\mathbf{I}_l$ as:

$$\mathfrak{D}[x,y] = d^\star(\mathbf{R}_i) \Leftarrow (x,y) \in \mathbf{R}_i \tag{5.1}$$



(*a*) (*b*) (*c*)

FIGURE 5.1: Superimpositions and semi-occlusions. (*a*) Left view of `Teddy`. The thick lines represent the coarse segmentation of the image, while the union of thick and thin lines represents the fine segmentation. (*b*) Absolute differences between the left and right stereo images for a left-to-right shift of 66 pixels and (*c*) 118 pixels. The region delineated by the white lines is in "best" superimposition with the right image of the stereo pair at 66 pixels, but the sum of absolute differences inside the whole region support is unlikely to constitute a pertinent superimposition cost.

where $d^\star(\mathbf{R}_i)$ corresponds to the regional disparity retained for and allotted to region $\mathbf{R}_i$. A simple method of computing this regional disparity is to minimise the sum of absolute differences of the brightness for the two superimposed images and across region $\mathbf{R}_i$ as follows:

$$d^\star(\mathbf{R}_i) = \arg\min_d \sum_{(x,y)\in\mathbf{R}_i} |\mathbf{I}_l[x,y] - \mathbf{I}_r[x-d,y]|$$

Note that with respect to the cost corresponding to the superposition of (x_i, y_i) in $\mathbf{I}_l$ with (x_j, y_i) in $\mathbf{I}_r$, as described by equation 2.2, this would amount to choosing the aggregation support $\mathcal{A}(x_i, x_j, y_i) = \{(x, y, x_i - x_j) \,|\, (x_i, y_i) \in \mathbf{R}_i \Rightarrow (x, y) \in \mathbf{R}_i\}$. Such an aggregation support, if used with the absolute differences of image brightness, proves inadequate if region $\mathbf{R}_i$ contains points occluded in $\mathbf{I}_r$. The reason behind this is clearly illustrated in figure 5.1(*b*), where the aggregation support overlaps two different regions of $\mathbf{I}_r$, despite correct registration. The methods proposed below enable computation of more pertinent regional disparities.

5.1.1 Gradient-based computation

Originally, the gradient-based computation of regional disparities was developed in order to register relatively homogeneous regions where there are slight brightness and colour discrepancies across the stereo pair. In this study case, the most important source of gradient information arises from the region contours. But as stated in section 2.3.2, special care has to be taken to properly interpret the disparities measured along the contours: should they be transferred to the region of interest, or to a neighbouring region, or both?

Assuming that $\mathbf{R}_i$ represents an object lying parallel to the image plane, we can then expect

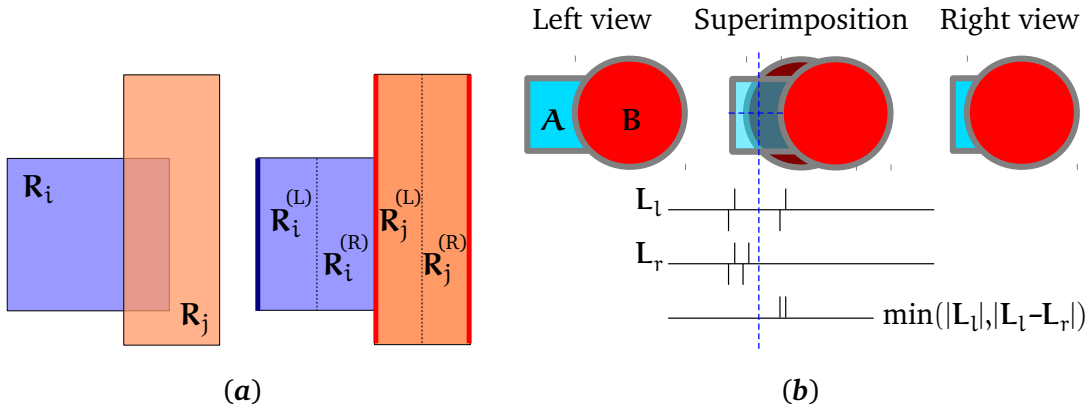


FIGURE 5.2: The logic behind the gradient-based computation of regional disparities. (*a*) Each region is split along its vertical skeleton into two left and right subregions. In this example, $\mathbf{R}_i$ is semi-occluded. The regional disparity of $\mathbf{R}_i^{(L)}$ is induced by the left vertical contour of $\mathbf{R}_i$, whereas the regional disparity of $\mathbf{R}_i^{(R)}$ is induced by the occlusion resulting from $\mathbf{R}_j^{(L)}$. Therefore, the regional disparity of $\mathbf{R}_i^{(R)}$ should be replaced by the one of $\mathbf{R}_i^{(L)}$. (*b*) There are extreme cases, where an occluding region would interfere in both subregions of the semi-occluded region. The asymmetric superimposition cost based on Laplacians limits the impact of this interference.

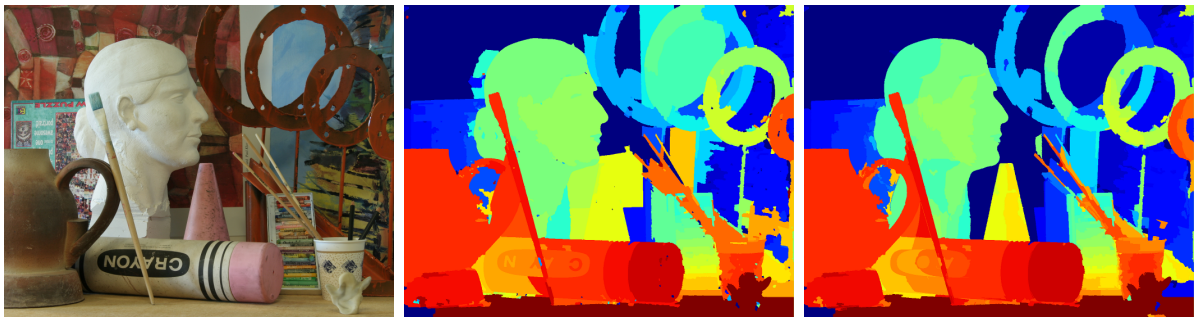
that if $\mathbf{R}_i$ is semi-occluded in the right image, there will be displacement d_1 for which the contours corresponding to the physical frontiers of $\mathbf{R}_i$ are in best superimposition, and a displacement $d_2 >> d_1$ for which its occlusion contours are in best superimposition. Furthermore, if d_2 corresponds to the actual disparity of an occluding contour, it should correspond to the regional disparity of a region adjacent to $\mathbf{R}_i$. Of course, if we are then left to choose between d_1 and d_2 for the regional disparity for $\mathbf{R}_i$, we should opt for the smallest measured displacement, i.e. d_1.

To proceed, every region $\mathbf{R}_i$ of the left image partition is split along its vertical skeleton into two subregions $\mathbf{R}_i^{(L)}$ and $\mathbf{R}_i^{(R)}$, as illustrated in figure 5.2(*a*). We then seek a regional disparity for each of these sub-regions. We could try a direct superimposition of the gradients corresponding to $\mathbf{I}_l$ and $\mathbf{I}_r$, but our experience showed that this typically leads to inaccuracies, in particular if the gradient is fairly thick. This problem can be bypassed by employing a signed Laplacian function $\mathbf{L}(\mathbf{I})$ instead which, given the function associated with input image $\mathbf{I}$, is computed as follows:

$$\mathbf{L}(\mathbf{I}) = \varepsilon_{\rightarrow}(\|\nabla \mathbf{I}\|) - \varepsilon_{\leftarrow}(\|\nabla \mathbf{I}\|) \tag{5.2}$$

The arrows $\leftarrow$ and $\rightarrow$ represent the directional structuring elements employed with the erosion operators. This Laplacian function has the advantage of highlighting the rising and descending edges of the gradient along the horizontal image axis. The regional disparity $d_G^\star(\widetilde{\mathbf{R}_i})$ of subregion $\widetilde{\mathbf{R}_i}$ associated with the gradient-based computation is then defined by equation 5.3.

$$d_G^\star(\widetilde{\mathbf{R}_i}) = \arg\min_d \sum_{(x,y)\in\widetilde{\mathbf{R}_i}} \min\{|\mathbf{L}_l[x,y]|, |\mathbf{L}_l[x,y] - \mathbf{L}_r[x-d,y]|\} \tag{5.3}$$



(*a*) (*b*) (*c*)

FIGURE 5.3: A first example of regional disparity map. (*a*) Input image `Art` originating from the Middlebury 2005 dataset. (*b*) The raw regional disparities obtained using the gradient-based computation *without* the analysis across subregions. As can be seen, the disparities of foreground objects are often inherited by the objects they occlude, or partly occlude, in the other image of the stereo pair. (*c*) Regional disparities displayed in each subregion of the partition *after rectification*. Semi-occluded regions are ultimately assigned perceptually relevant disparities, though the problem remains unsolved for those regions fully surrounded by occluding contours, such as the gap between the brush, the chalk and the jar.

with $\mathbf{L}_l$ and $\mathbf{L}_r$ representing the short-cut notations of $\mathbf{L}(\mathbf{I}_l)$ and $\mathbf{L}(\mathbf{I}_r)$ respectively. The absolute difference between the Laplacians ensures that the gradients are correctly superimposed, whilst the minimum with $|\mathbf{L}_l|$ is enforced, so that the occluding contours do not interfere with the aggregation costs, as shown in figure 5.2(*b*). The regional disparity of $\mathbf{R}_i$ is finally estimated directly as

$$d_G^\star(\mathbf{R}_i) = \min\left\{ d_G^\star\left(\mathbf{R}_i^{(L)}\right), d_G^\star\left(\mathbf{R}_i^{(R)}\right)\right\}$$

In some cases though, it is useful to compute the regional disparity map associated with the partition containing the subregions. Indeed, when two subregions share the same parent region, the map will be coherent if both subregions are allocated different regional disparities, in particular if the parent region is tilted around some vertical axis. However, major differences in regional disparities could indicate that the highest regional disparity is the product of an occlusion. In order to establish whether the subregion assigned the highest regional disparity has inherited from the regional disparity of an occluded region, it suffices to find an adjacent subregion which has been allocated a similar regional disparity. In the affirmative case, the regional disparity of the subregion under consideration is finally replaced with the regional disparity of the other subregion belonging to the same parent region. Figure 5.3 shows the efficacy of this rectification method on one example from the Middlebury database.

The second method presented in this section investigates the possibilities offered by a change of aggregation support and is more robust than the method derived from gradient-based regional disparities, when processing images which have no difference in colour.

5.1.2 Lightness-based computation

In section 2.1.2, we focused our attention on using regional intersections between the partitions of $\mathbf{I}_l$ and $\mathbf{I}_r$ to aggregate superimposition costs. We shall now reuse this idea to define the *illumination* cost of superimposing $\mathbf{R}_i$ with $\mathbf{R}'_j$ for a left-to-right displacement of d pixels as:

$$c_{\text{Illum}}(d, \mathbf{R}_i, \mathbf{R}'_j) = \frac{1}{|\mathbf{R}_i \cap \mathbf{R}'^{(d)}_j|} \sum_{(x,y)\in \mathbf{R}_i \cap \mathbf{R}'^{(d)}_j} |\mathbf{I}_l[x,y] - \mathbf{I}_r[x-d,y]| \tag{5.4}$$

If the input images are composed of multiple channels, it is possible to take the mean average of the costs computed for each as the final illumination cost. The cost computed in equation 5.4 will unfortunately lack relevance when the size of the intersection is very small compared to the size of the original regions. The significance of an intersection between two regions $\mathbf{R}_i$ and $\mathbf{R}'^{(d)}_j$ can be measured by the Jaccard distance [Jaccard, 1901], as follows:

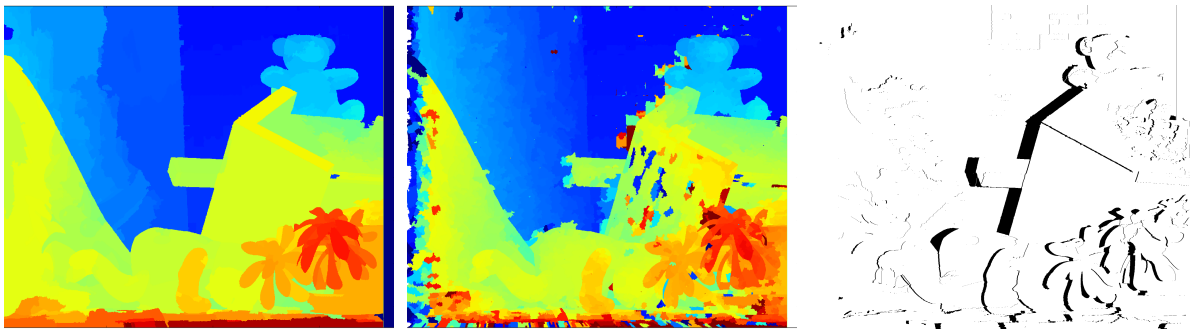
$$c_{\text{Jaccard}}(d, \mathbf{R}_i, \mathbf{R}'_j) = 1 - \frac{|\mathbf{R}_i \cap \mathbf{R}'^{(d)}_j|}{|\mathbf{R}_i \cup \mathbf{R}'^{(d)}_j|} \tag{5.5}$$

In fact, the Jaccard distance plays the role of a *coverage* cost between two superimposed regions. The asymmetrical version of this coverage cost, which is obtained by replacing $|\mathbf{R}_i \cup \mathbf{R}'^{(d)}_j|$ with $|\mathbf{R}_i|$ in equation 5.5, is useful if the segmentation of $\mathbf{I}_r$ is coarser than that of $\mathbf{I}_l$.

We now have two criteria based on image contents and shape similarities, which can be combined to compute regional disparities. We simply propose to discredit the superimposition of $\mathbf{R}_i$ with $\mathbf{R}'^{(d)}_j$ if the asymmetrical Jaccard distance between $\mathbf{R}_i$ and $\mathbf{R}'^{(d)}_j$ is above $\tau = 0.75$, resulting in less than 25% of the region of $\mathbf{R}_i$ being superimposed with the candidate matching region $\mathbf{R}'_j$ when shifting the right image from left to right, by a magnitude of d pixels. The regional disparity $d^\star_L(\mathbf{R}_i)$ associated with the lightness-based computation is then defined by equation 5.6.

$$\begin{aligned} \tilde{c}_{\text{Illum}}(d, \mathbf{R}_i, \mathbf{R}'_j) &= \begin{cases} c_{\text{Illum}}(d, \mathbf{R}_i, \mathbf{R}'_j) & \text{if } c_{\text{Jaccard}}(d, \mathbf{R}_i, \mathbf{R}'_j) \leqslant \tau \\ +\infty & \text{otherwise} \end{cases} \\ d^\star_L(\mathbf{R}_i) &= \arg\min_d \left\{ \min_{\mathbf{R}'_j} \tilde{c}_{\text{Illum}}(d, \mathbf{R}_i, \mathbf{R}'_j) \right\} \end{aligned} \tag{5.6}$$

As figure 5.4(***a***) shows, these regional disparities are also particularly resistant to occlusions. The two assumptions of this alternative are that corresponding pixels between the two stereo images have the same lightness and that semi-occlusions occurring in the right image do not occlude more than 75% of the region of interest in the left image.



(*a*) (*b*) (*c*)

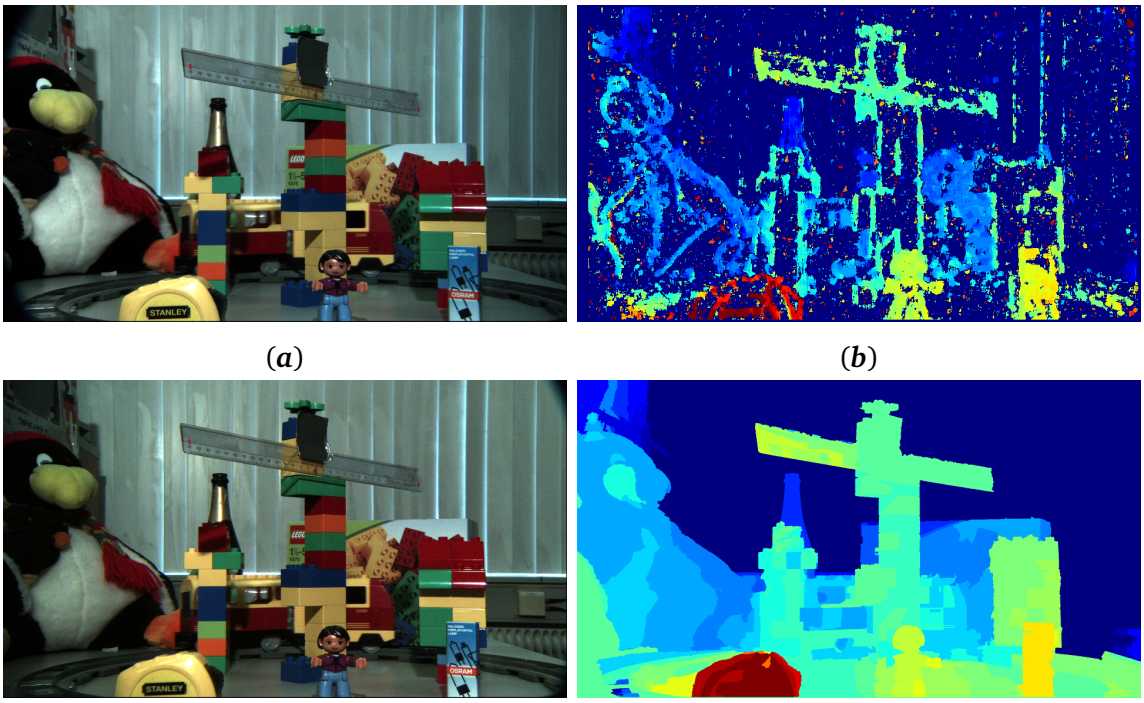
FIGURE 5.4: Regional disparities and segmentation hierarchies. (*a*) The regional disparity map for the coarse partition of `Teddy` using the lightness-based computation is, at a global observation scale, consistent with respect to depth perception. (*b*) The same regional disparities computed for the finer partition are more accurate across tilted regions with respect to the image plane. However, the errors caused by the total occlusion of small regions and by the non-textured areas are perceptually disturbing. (*c*) Using the coarse regional disparity map shown in (*a*), it is possible to generate a binary mask to estimate the image areas subject to the occlusion phenomenon. Here, only object-type occlusions are shown in black.

5.1.3 Properties of regional disparity maps

The regional disparity maps generated by equation 5.1 come with a wealth of positive features, although some requirements must be fulfilled to ensure the quality of the end result. In this last subsection on the measurements related to regional disparities, we draw an objective analysis of the measure.

Underlying assumptions In order to compute the regional disparities of a coarse image partition, it has been assumed that the regions could be registered across the stereo pair by means of translations. Theoretically, this is only true if the regions correspond to planar objects parallel to the image plane, i.e. if all objects are *fronto-parallel* to the camera. Nonetheless, when the baseline between the two sensors acquiring the pair of stereo images is low, the geometrical distortions between the image regions are less important than the distortions, which result when using a wide baseline. Thus translational registration model still produces stable measurements. The same applies to the wide baseline case if regions are not excessively tilted.

Perceptual features and accuracy Regional disparity maps are strongly related to the segmentations from which they originate. This encourages the generation of sharp disparity discontinuities at regional boundaries. Furthermore, regional disparity maps are not subject to the fattening effect observed in a large number of pixel-based approaches and the depth ordering between different objects in the scene is therefore consistently revealed. Another advantageous feature of regional disparity maps resides in their ability to handle semi-occlusions: indeed, any



(*a*) (*b*)

(*c*) (*d*)

FIGURE 5.5: Regional disparities and microstereopsis. (*a*) and (*c*) represent the left and right views of the stereo pair respectively. (*b*) The disparity map obtained using the semi-global matching algorithm [Hirschmüller, 2008]. Disparities are captured well around contours, but the main source of difficulty arises from the homogeneous regions of the scene and the fattening effect can be observed around every contour, especially for the foreground objects, such as the figurine, the tape roll, the little box and the ruler. (*d*) The regional disparity map obtained using the gradient-based computation.

pixel occluded in one image of the stereo pair inherits the regional disparity of the region to which it belongs. Not only does this make the disparity of the occluded pixel relevant to those which belong to the same region, but it also eliminates any need for an additional diffusion or procedure to estimate the disparities across the region's occluded areas. However, since the pixels of any given tilted region all receive the same disparity, regional disparity maps will always constitute gross estimations of the real disparity functions and therefore cannot be accurate. Fortunately, it is possible to refine the disparity maps by using regional disparities at a finer degree of segmentation, i.e. with more over-segmentation.

Hierarchical characteristics The computation of regional disparities driven by a finer partition typically yields regional disparity maps, which are more accurate than those observed at the coarse level of segmentation. But using smaller regions comes at a price: it is likely that a greater number of regions will be totally occluded, making their associated disparity measures completely erroneous, and that across homogeneous regions, a greater number of ambiguities will arise. This can be observed in figure 5.4(*b*). In this scenario, the coarse regional disparity map associated with the same image, though inaccurate, may constitute a very pertinent a priori assumption to

constrain the refinement of regional disparities at a finer level of a *hierarchical* segmentation. The first thing to note is that the regional disparity of a fine region should never be at a great distance from the regional disparity of the gross region of which it is part. The second, is that it is easy, using the coarse regional disparity map, to derive the image areas subject to object occlusions. These can be gathered in an occlusion binary mask $\mathbf{M}_{\text{occl.}}$, of the same size as the coarse regional disparity map $\mathfrak{D}$, such that:

$$S(\mathbf{M}_{\text{occl.}}) = \left\{ (x,y) \mid \underbrace{(x - \mathfrak{D}[x,y] < 0)}_{\text{border occlusion}} \vee \underbrace{\left(\exists\, x' > x \mid x' - \mathfrak{D}[x',y] = x - \mathfrak{D}[x,y]\right)}_{\text{object occlusion}} \right\} \tag{5.7}$$

For each region of the fine partition, it is therefore possible to measure what proportion is occluded and to determine whether or not its associated disparity measure is relevant. Refinements across partitions related by a hierarchy will be treated as part of our study on disparity *estimation*, in section 7.2.

5.2 Regional aggregations and point disparities

This second section presents an alternative mechanism which, given a pair of stereo images, produces accurate disparity measurements. As was the case with regional disparities, the intersections of the left and right image segmentations will be used to constrain the aggregation of superimposition costs. Furthermore, the detection of disparities induced by occluding contours will again prove useful during the filtering of bad disparity measures. However, the aggregation supports will be defined on a pixel basis in order to allow variations of disparities across tilted objects. Therefore, unlike regional disparities, the fronto-parallel assumption will no longer hold.

The purpose of the aggregation phase in stereo image analysis is to filter a disparity space volume (cf. section 2.1.4) in such a way that the minimum superimposition cost observed for a given pixel will occur for a meaningful displacement, corresponding to the measured disparity. Because the filtering essentially aims to smooth or average the costs in a particular neighbourhood, the term *diffusion* is sometimes employed in place of *aggregation*, especially when referring to a process which updates the cost values locally and iteratively. Experiments have already dealt with different diffusion costs in stereo. For example, the diffusion model chosen by [Scharstein and Szeliski, 1998] would, if no constraint were applied, amount to performing a Gaussian convolution on each disparity plane of the cost volume. As they show in their article, one advantage of iterative schemes over straightforward filtering operations lies in the ability to easily interrupt the diffusion based on local stopping criteria, and to dynamically adapt the diffusion strength. In [De-Maeztu et al., 2012], this strength is related to a geodesic colour distance, which separates the pixel undergoing the cost update from the distant pixel contributing

to this update. However, in [Cigla and Alatan, 2013], the more that high frequency areas are traversed by a diffusion path, the less significant becomes the contribution of the cost conveyed along this path.

2D diffusion In order to introduce our cost diffusion algorithm, let us start with a relatively straightforward example taking place in 2D space, without any constraint. Suppose we want to perform a diffusion having a scope of n pixels, which transforms an image $\mathbf{I}$ into $\tilde{\mathbf{I}}_n^{(L)}$, such that

$$\tilde{\mathbf{I}}_n^{(L)}[x,y] = \frac{1}{n+1}\sum_{i=0}^{n}\mathbf{I}[x-i,y]$$

By developing this equation and setting $i = j+1$, one obtains:

$$\begin{aligned}\tilde{\mathbf{I}}_n^{(L)}[x,y] &= \frac{1}{n+1}\left(\mathbf{I}[x,y]+\sum_{i=1}^{n}\mathbf{I}[x-i,y]\right)\\ &= \frac{1}{n+1}\left(\mathbf{I}[x,y]+\sum_{j=0}^{n-1}\mathbf{I}[(x-1)-j,y]\right)\\ &= \frac{1}{n+1}\left(\mathbf{I}[x,y]+n\cdot\tilde{\mathbf{I}}_{n-1}^{(L)}[x-1,y]\right)\end{aligned}$$

Observing that $\tilde{\mathbf{I}}_{n-1}^{(L)}[x-1,y]$ can be written as $\left(\varepsilon_{\mathbf{B}_L}\left(\tilde{\mathbf{I}}_{n-1}^{(L)}\right)\right)[x,y]$, where $S(\mathbf{B}_L)=\{(-1,0)\}$ represents the set of points parametrising a directional structuring element oriented towards the left-hand side of the image and *not* containing the centre point, we conclude that:

$$\tilde{\mathbf{I}}_n^{(L)} = \frac{1}{n+1}\left(\mathbf{I}+n\cdot\varepsilon_{\mathbf{B}_L}\left(\tilde{\mathbf{I}}_{n-1}^{(L)}\right)\right),\ \forall n\geqslant 1$$

Now, let $S(\mathbf{B}_R)=\{(+1,0)\}$, $S(\mathbf{B}_T)=\{(0,-1)\}$ and $S(\mathbf{B}_B)=\{(0,+1)\}$ represent the sets of points of structuring elements $\mathbf{B}_R$, $\mathbf{B}_T$ and $\mathbf{B}_B$ inducing the unitary translations towards the left, the bottom and the top respectively of the image plane under consideration, when used with the erosion operator. A horizontal moving average filter of size $2n+1$ on $\mathbf{I}$ yields as output:

$$\mathbf{I}_{\mathcal{H}} = \frac{1}{2n+1}\left(\mathbf{I}+n\cdot\varepsilon_{\mathbf{B}_L}\left(\tilde{\mathbf{I}}_{n-1}^{(L)}\right)+n\cdot\varepsilon_{\mathbf{B}_R}\left(\tilde{\mathbf{I}}_{n-1}^{(R)}\right)\right)$$

And since the multi-dimensional moving average filter is linearly separable [Szeliski, 2011], we can finally deduce the 2D moving average of image $\mathbf{I}$ as:

$$\mathbf{I}_{\mathcal{A}} = \frac{1}{2n+1}\left(\mathbf{I}+n\cdot\varepsilon_{\mathbf{B}_T}\left(\widetilde{\mathbf{I}_{\mathcal{H}}}_{n-1}^{(T)}\right)+n\cdot\varepsilon_{\mathbf{B}_B}\left(\widetilde{\mathbf{I}_{\mathcal{H}}}_{n-1}^{(B)}\right)\right) \tag{5.8}$$

This derivation shows that it is possible to build an iterative diffusion model which ultimately results in a simple moving average filter on a plane, if no constraint intervenes. Of course, the diffusion will not be applied to an image, but to a 3D volume, containing the stereo superimpo-

sition costs. [Cigla and Alatan, 2013] chose a very similar diffusion scheme with permeability constraints and made their algorithm operate independently on each disparity plane. However, this latter feature is precisely what we wish to avoid in this second approach to the measurement of disparities, since it favours fronto-parallel configurations to the detriment of tilted configurations. Besides, in a scenario where both $n \to +\infty$ and the diffusion is impeded at the regional boundaries, the cost allocated to an arbitrary pixel would tend towards the average cost of the pixels belonging to the same region. This recalls the cost computed for the regional disparities, which we know to be inaccurate.

3D diffusion Our cost diffusion scheme therefore needs to tolerate 3D diffusion paths. More precisely, we can keep the 2D aggregation scheme based on the moving average filter, but this time, we should consider the possibility that the 2D surface we would like to span with our diffusion algorithm, may be twisted inside the disparity space volume. Therefore, a unitary variation of disparity may occur every time the diffusion path progresses along a particular direction of the image plane. It is impossible to know in advance which variation of disparity will be favoured at a given step of the diffusion. However, we know how to compute the cost of the path which best warps both stereo images along the left-to-right direction, according to equation 2.6. By slightly adapting this equation, we can handle other warping directions with respect to the image plane and impose that the path should systematically evolve in the chosen direction, i.e. that no two points of the warping path will project onto the same image plane coordinates. Furthermore, we can choose the maximum length n of the warping path, which is of interest to us. Assuming that the direction of propagation is $\mathbf{d} = (d_x, d_y) \neq \mathbf{0}$ with respect to the image plane, the 3D diffusion transforming volume $\mathbf{D} = \tilde{\mathbf{D}}_0^{(\mathbf{d})}$ into $\tilde{\mathbf{D}}_n^{(\mathbf{d})}$ then becomes:

$$\tilde{\mathbf{D}}_n^{(\mathbf{d})} = \frac{1}{n+1}\left(\mathbf{D} + n \cdot \min\left\{\varepsilon_{\mathbf{B}_0}\left(\tilde{\mathbf{D}}_{n-1}^{(\mathbf{d})}\right), \varepsilon_{\mathbf{B}_1}\left(\tilde{\mathbf{D}}_{n-1}^{(\mathbf{d})}\right) + \xi\right\}\right), \forall n \geqslant 1 \tag{5.9}$$

with $S(\mathbf{B}_0) = \{(d_x, d_y, 0)\}$ and $S(\mathbf{B}_1) = \{(d_x, d_y, -1), (d_x, d_y, +1)\}$ representing the sets of points for the fronto-parallel and tilted structuring elements respectively. The impact of the regularisation term ξ on the diffusion mechanism will be discussed later in this section. We now have all the necessary ingredients required to define our cost diffusion algorithm. All that remains for us to do is to integrate the segmentations of the left and right images as the main diffusion constraint.

5.2.1 Cost diffusion algorithm

The cost diffusion algorithm requires as input a disparity space volume $\mathbf{D}$ representative of the stereo image superimpositions, the partitions $\mathcal{L}_l$ and $\mathcal{L}_r$ of the left and right images respectively, as well as the maximum scope of the cost diffusion, n. First we provide some particulars about

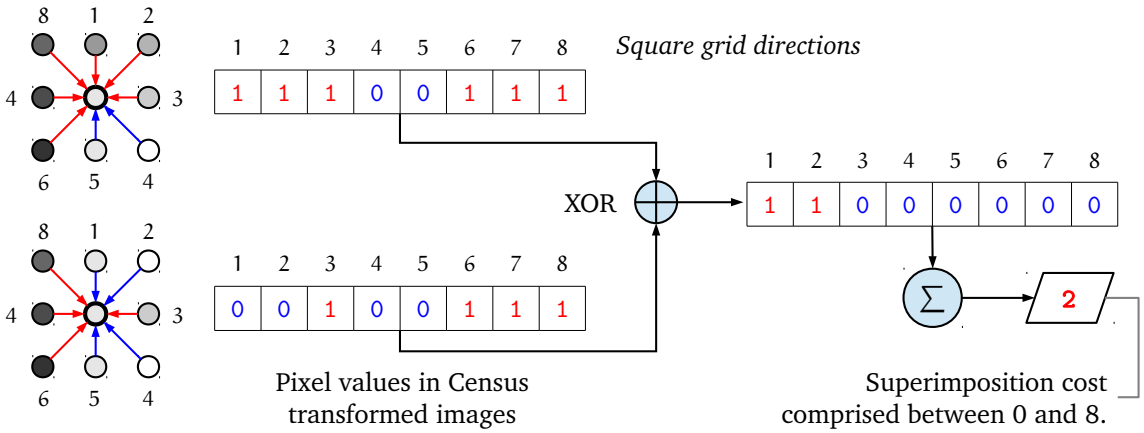


FIGURE 5.6: Binary representation of the values resulting from the Census transformation with respect to the two pixels circled in bold, and illustration of the mechanism allowing the computation of their superimposition cost, based on the Hamming distance between the two binary sequences associated with their values in the Census transformed images.

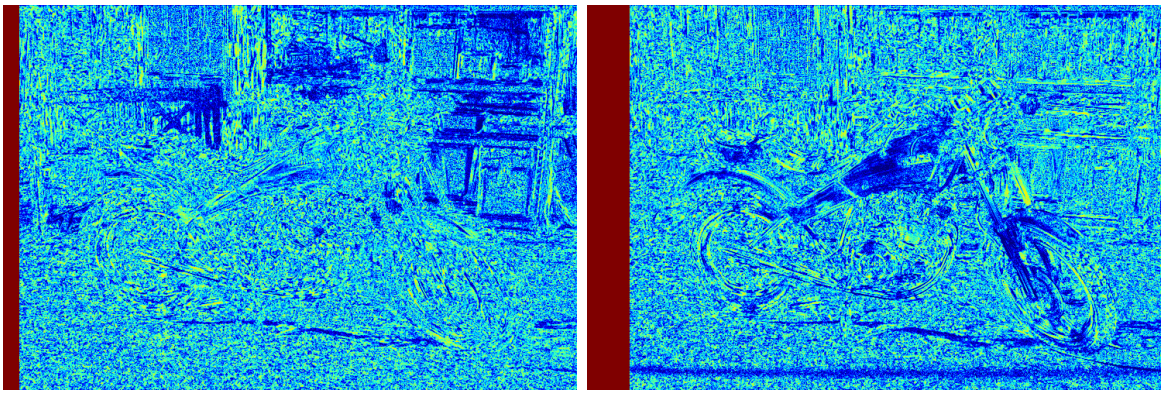
these parameters which we follow with details of our algorithm.

DSV specifications Each entry of the disparity space volume $\mathbf{D}$ corresponds to a superimposition cost between one pixel of the left view and one pixel of the right view. In chapter 2, we defined this cost as the absolute difference between the lightness values attributed to each pixel of the pair under consideration. Of course, this is a bad choice when illumination discrepancies manifest between the two images of the stereo pair. In this algorithm, we derive each superimposition cost from the Census transformed stereo images [Zabih and Woodfill, 1994]. Applying the Census transformation to a greyscale image $\mathbf{I}$ consists of mapping each pixel of $\mathbf{I}$ into a binary code composed of 8-bits. With respect to the square grid, a direction corresponds to each bit of the binary code. A bit is activated for a given pixel, if and only if the neighbour pixel in the direction associated with the bit being considered, has a lightness value smaller than that of the given pixel. More formally, the Census transformation maps an image $\mathbf{I}$ into another image of identical dimensions $\mathtt{Census}(\mathbf{I})$, according to equation 5.10:

$$\mathtt{Census}(\mathbf{I})[x,y] = \sum_{i=0}^{7} \mathtt{T}(\mathbf{I} - \varepsilon_{\mathbf{B}_i}(\mathbf{I}))[x,y] \cdot 2^i \tag{5.10}$$

where $\mathtt{T}$ represents a binary indicator function, such that $\mathtt{T}(\mathbf{I})[x,y] = 1 \Leftrightarrow \mathbf{I}[x,y] > 0$ and $\mathbf{B}_i$ corresponds to the i-th directional structuring element defined on the square grid. Figure 5.6 illustrates that particular transformation.

Once the Census transformed images are computed, it is possible to compute a dissimilarity cost between any pixel of the left view and any pixel of the right view, by considering the Hamming distance between their associated binary codes. Each entry of the *normalised* DSV is ultimately



(*a*) (*b*)

FIGURE 5.7: Visualisation of the disparity space volume computed according to equation 5.11 for the `Motorcycle` stereo pair. Volume slices obtained for (*a*) disparity $d = 21$ pixels and (*b*) disparity $d = 60$ pixels. Input image dimensions: 741×497 pixels.

computed as:

$$\mathbf{D}[x, y, d] = \frac{1}{8} \cdot \sum_{i=0}^{7} \left\lfloor \frac{\texttt{Census}(\mathbf{I}_l)[x, y] \oplus \texttt{Census}(\mathbf{I}_r)[x - d, y]}{2^i} \right\rfloor \mod 2 \tag{5.11}$$

where $\oplus$ denotes the XOR operator between the binary sequences associated with two integers. In the case of colour images, three DSVs are computed, one for each image channel according to equation 5.11, and $\mathbf{D}$ is equal to their mean average.

Since the Census transformation is only sensitive to the directions in which the lightness is lower than that of the pixel under consideration, the superimposition costs computed according to equation 5.11 remain relevant when the illumination discrepancies manifesting between the stereo images are due to a strictly increasing transformation function. Furthermore a null superimposition cost between two pixels indicates that the variations of lightness within their direct neighbourhood are similar, which suggests that the two pixels effectively match with more certainty than would a null superimposition cost which is computed as the absolute difference between two pixel lightness values.

Segmentation characteristics The partitions $\mathcal{L}_l$ and $\mathcal{L}_r$ are generated according to the over-segmentation algorithm presented in section 4.3. Each of these partitions is in fact an image where each pixel maps to the label which uniquely identifies the region to which it belongs. We call $\mathcal{L}_l[x, y]$ the label of pixel (x, y) in partition $\mathcal{L}_l$, and $\mathcal{L}_r[x, y]$ the label of pixel (x, y) in partition $\mathcal{L}_r$. If we superimpose these two partitions, we can compute their intersection. The intersection of two partitions is a partition, where each pixel has a label which uniquely identifies the regions where the pixel is included with respect to the two input partitions. We can then generate a volume containing the intersections of $\mathcal{L}_l$ and $\mathcal{L}_r$ for different left-to-right displacements applied

$\mathcal{L}_V$								
$\varepsilon_{\mathbf{B}}\mathcal{L}_V$								
$\mathbf{D}$	7	7	8	4	3	1	1	0
$\mathcal{D}_{\mathbf{B}}$	18	19	20	**0**	1	2	3	4
$\mathbf{D}_{\mathrm{OUT}}$ @ t=0	7	7	8	4	3	1	1	0
$\mathbf{D}_{\mathrm{OUT}}$ @ t=1	?	14	15	4	7	4	2	1
$\mathbf{D}_{\mathrm{OUT}}$ @ t=2	?	?	22	4	7	8	5	2
$\mathbf{D}_{\mathrm{OUT}}$ @ t=3	?	?	?	4	7	8	9	5
$\mathbf{D}_{\mathrm{OUT}}$ @ t=4	?	?	?	4	7	8	9	9
$\mathbf{D}_{\mathrm{OUT}} \div (\min\{4, \mathcal{D}_{\mathbf{B}}\} + 1)$	?	?	?	**4**	**3.5**	**2.7**	**2.3**	**1.8**

FIGURE 5.8: Execution sample of the DIRECTIONALDIFFUSION function of algorithm 5.1 illustrating how the distance function $\mathcal{D}_{\mathbf{B}}$, derived from the volume $\mathcal{L}_V$ encoding the superimpositions between the left and right image segmentations, constrains the diffusion so that no costs arising from different regional intersections can combine.

to the right image partition $\mathcal{L}_r$. We call this volume $\mathcal{L}_V$ and we express it as:

$$\mathcal{L}_V[x, y, d] = \mathcal{L}_l[x, y] + \mathcal{L}_r[x - d, y] \cdot \max_{x,y} \mathcal{L}_l[x, y]$$

If we propagate the costs along the direction opposite to that indicated by a translational structuring element $\mathbf{B}_i$, such that the scope of the propagation equals $n = 1$ pixel only, then the region from which the cost propagated at voxel (x, y, d) originates is simply given by $\varepsilon_{\mathbf{B}_i}(\mathcal{L}_V)[x, y, d]$. The voxels where costs from different regions would be combined if the diffusion were happening without constraint, are identified by the set

$$\mathcal{S}_{\mathbf{B}_i} = \{(x, y, d) \mid \mathcal{L}_V[x, y, d] \neq \varepsilon_{\mathbf{B}_i}(\mathcal{L}_V)[x, y, d]\}$$

Of course, if we use the diffusion model represented by equation 5.9, the set containing the voxels where the costs of different regions have the potential to combine, is defined by $\mathcal{S}_{\mathbf{B}}$ for $S(\mathbf{B}) = \mathbf{B}_0 \cup \mathbf{B}_1$.

Diffusion constrained by distance functions We call $\mathcal{D}_{\mathbf{B}}$ the distance function associated with the binary volume for which the deactivated voxels are all those belonging the set $\mathcal{S}_{\mathbf{B}}$, i.e. $\mathcal{D}_{\mathbf{B}}[x, y, d] = 0 \Leftrightarrow (x, y, d) \in \mathcal{S}_{\mathbf{B}}$. $\mathcal{D}_{\mathbf{B}}$ may be computed by means of successive erosions on this binary volume, according to the algorithm presented in section 3.1.1, but replacing the isotropic structuring element with structuring element $\mathbf{B}$. Algorithm 5.1, which implements our 3D cost diffusion method, uses this distance function at line 9, in order to constrain the directional diffusions driven by structuring element $\mathbf{B}$. The voxels for which the costs of different regions

Algorithm 5.1 3D diffusion of superimposition costs

function DIRECTIONALDIFFUSION($\mathbf{D}$, n, ξ, (d_x, d_y))
  $t \leftarrow 0$
  $\mathbf{D}_{\mathsf{OUT}} \leftarrow \mathbf{D}$
  $\mathbf{B}_0 \leftarrow$ Structuring element defined by $S(\mathbf{B}_0) = \{(d_x, d_y, 0)\}$
  $\mathbf{B}_1 \leftarrow$ Structuring element defined by $S(\mathbf{B}_1) = \{(d_x, d_y, -1), (d_x, d_y, +1)\}$
  $\mathbf{B} \leftarrow \mathbf{B}_0 \cup \mathbf{B}_1$
  while $t < n$ **do**
    $\mathbf{D}_{\mathsf{UPD}} \leftarrow \mathbf{D} + \min\{\varepsilon_{\mathbf{D}_0}(\mathbf{D}_{\mathsf{OUT}}), \varepsilon_{\mathbf{D}_1}(\mathbf{D}_{\mathsf{OUT}}) + \xi\}$
    $\mathbf{D}_{\mathsf{SEL}} \leftarrow$ Binary volume indicating voxels where $\mathcal{D}_{\mathbf{B}} > t$
    $\mathbf{D}_{\mathsf{OUT}} \leftarrow \mathbf{D}_{\mathsf{UPD}} \cdot \mathbf{D}_{\mathsf{SEL}} + \mathbf{D}_{\mathsf{OUT}} \cdot (1 - \mathbf{D}_{\mathsf{SEL}})$
    $t \leftarrow t + 1$
  return $\mathbf{D}_{\mathsf{OUT}}$

function DIFFUSECOSTS($\mathbf{D}$, n, ξ)
  Initialisation of structuring elements
  $\mathbf{B}_L \leftarrow$ Structuring element defined by $S(\mathbf{B}_L) = \bigcup_{d_z \in \{-1,0,+1\}} \{(-1, 0, d_z)\}$
  $\mathbf{B}_R \leftarrow$ Structuring element defined by $S(\mathbf{B}_R) = \bigcup_{d_z \in \{-1,0,+1\}} \{(+1, 0, d_z)\}$
  $\mathbf{B}_T \leftarrow$ Structuring element defined by $S(\mathbf{B}_T) = \bigcup_{d_z \in \{-1,0,+1\}} \{(0, -1, d_z)\}$
  $\mathbf{B}_B \leftarrow$ Structuring element defined by $S(\mathbf{B}_B) = \bigcup_{d_z \in \{-1,0,+1\}} \{(0, +1, d_z)\}$
  Diffusion algorithm
  $\mathbf{D}_{\mathsf{XL}} \leftarrow$ DIRECTIONALDIFFUSION($\mathbf{D}$, n, ξ, $(-1, 0)$)
  $\mathbf{D}_{\mathsf{XR}} \leftarrow$ DIRECTIONALDIFFUSION($\mathbf{D}$, n, ξ, $(+1, 0)$)
  $\mathbf{D}_{\mathsf{X}} \leftarrow (\mathbf{D}_{\mathsf{XL}} + \mathbf{D}_{\mathsf{XR}} - \mathbf{D}) \div (\min\{n, \mathcal{D}_{\mathbf{B}_L}\} + \min\{n, \mathcal{D}_{\mathbf{B}_R}\} + 1)$
  $\mathbf{D}_{\mathsf{YT}} \leftarrow$ DIRECTIONALDIFFUSION($\mathbf{D}_{\mathsf{X}}$, n, ξ, $(0, -1)$)
  $\mathbf{D}_{\mathsf{YB}} \leftarrow$ DIRECTIONALDIFFUSION($\mathbf{D}_{\mathsf{X}}$, n, ξ, $(0, +1)$)
  $\mathbf{D}_{\mathsf{Y}} \leftarrow (\mathbf{D}_{\mathsf{YT}} + \mathbf{D}_{\mathsf{YB}} - \mathbf{D}_{\mathsf{X}}) \div (\min\{n, \mathcal{D}_{\mathbf{B}_T}\} + \min\{n, \mathcal{D}_{\mathbf{B}_B}\} + 1)$
  return $\mathbf{D}_{\mathsf{Y}}$

would merge at the first iteration of the diffusion, are those where $\mathcal{D}_{\mathbf{B}} = 0$. These appear as deactivated in $\mathbf{D}_{\mathsf{SEL}}$, which prevents their cost update at line 10. At the start of iteration $t = 1$, their costs in $\mathbf{D}_{\mathsf{OUT}}$ are identical to those of $\mathbf{D}$. However, their costs have been propagated to some neighbour voxels in the previous iteration, and diffusing them again would add influence to the initial superimposition costs close to the regional boundaries, which would unbalance the weights of each initial cost. Therefore, at $t = 1$, the voxels where $\mathcal{D}_{\mathbf{B}} = 1$ are also deactivated in $\mathbf{D}_{\mathsf{SEL}}$, and their cost update is definitively stopped. The process continues in this manner until the maximum scope of the diffusion is attained. Figure 5.8 illustrates the execution of such a directional diffusion using a one-dimensional example.

Algorithm's features Algorithm 5.1 is composed of two functions. The first performs the 3D directional diffusion of the costs *constrained* by the superimpositions of the left and right image partitions. The update rule for the directional diffusion at line 8 is based on equation 5.9, which resorts to directional warping. In the second function, the costs are initially diffused separately along the left and right directions of the image plane. They are then combined into a new cost volume, which is in turn diffused with respect to the two vertical directions of the image plane. The combination of the two vertical diffusions ultimately yields the output cost volume. This is similar to the aggregation scheme of equation 5.8. Note that, due to the boundary constraint, the normalisation factors may vary for each voxel. They depend on distances which separate the voxel under consideration from the segmentation boundaries, according to the four directions tested. Furthemore, by construction, more weight is given to the diffusion coming from the direction where the segmentation boundary is farthest from the voxel being considered.

Resulting sparse disparity maps Let $\mathfrak{D}^{(\text{init})}$ be the *sparse* disparity map resulting from the diffused cost volume, say $\tilde{\mathbf{D}}$, produced by algorithm 5.1. A sparse disparity map contains pixels with or without a disparity value. Pixels which are not allocated a disparity value are those for which the measure is ambiguous, for instance across homogeneous regions, or for which the measure is simply not feasible because of occlusions. In chapter 2, we saw how to detect such configurations using the cross-checking criterion expressed by equation 2.7. Therefore, we define $\mathfrak{D}^{(\text{init})}$ as

$$\mathfrak{D}^{(\text{init})}[x, y] = \arg\min_{d} \tilde{\mathbf{D}}[x, y, d] \tag{5.12}$$

for any pixel (x, y) which satisfies the cross-checking criterion when allocated disparity d according to the disparity space volume $\tilde{\mathbf{D}}$. The disparity remains undefined for any other pixels. Some examples of disparity maps are provided in figures 5.9 and 5.10.

Parameters' impact Now, by referring to the aforementioned figures, we can observe the impact of the diffusion scope and regularisation parameters n and ξ on the sparse disparity maps deduced from the diffused cost volumes. For each disparity map, we also measured the ratio of pixels allocated a disparity measure in excess of one unit above or below the ground truth

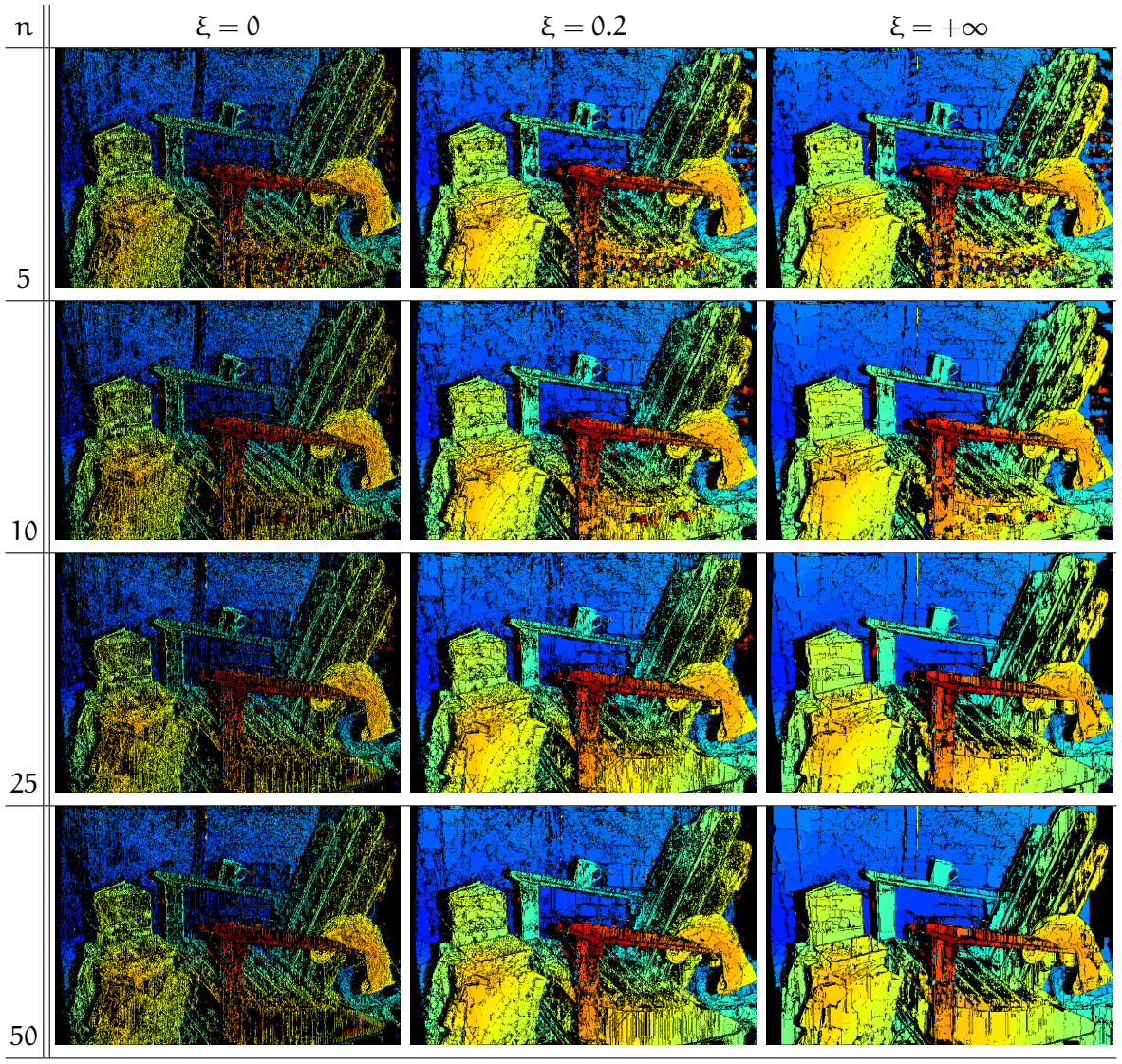
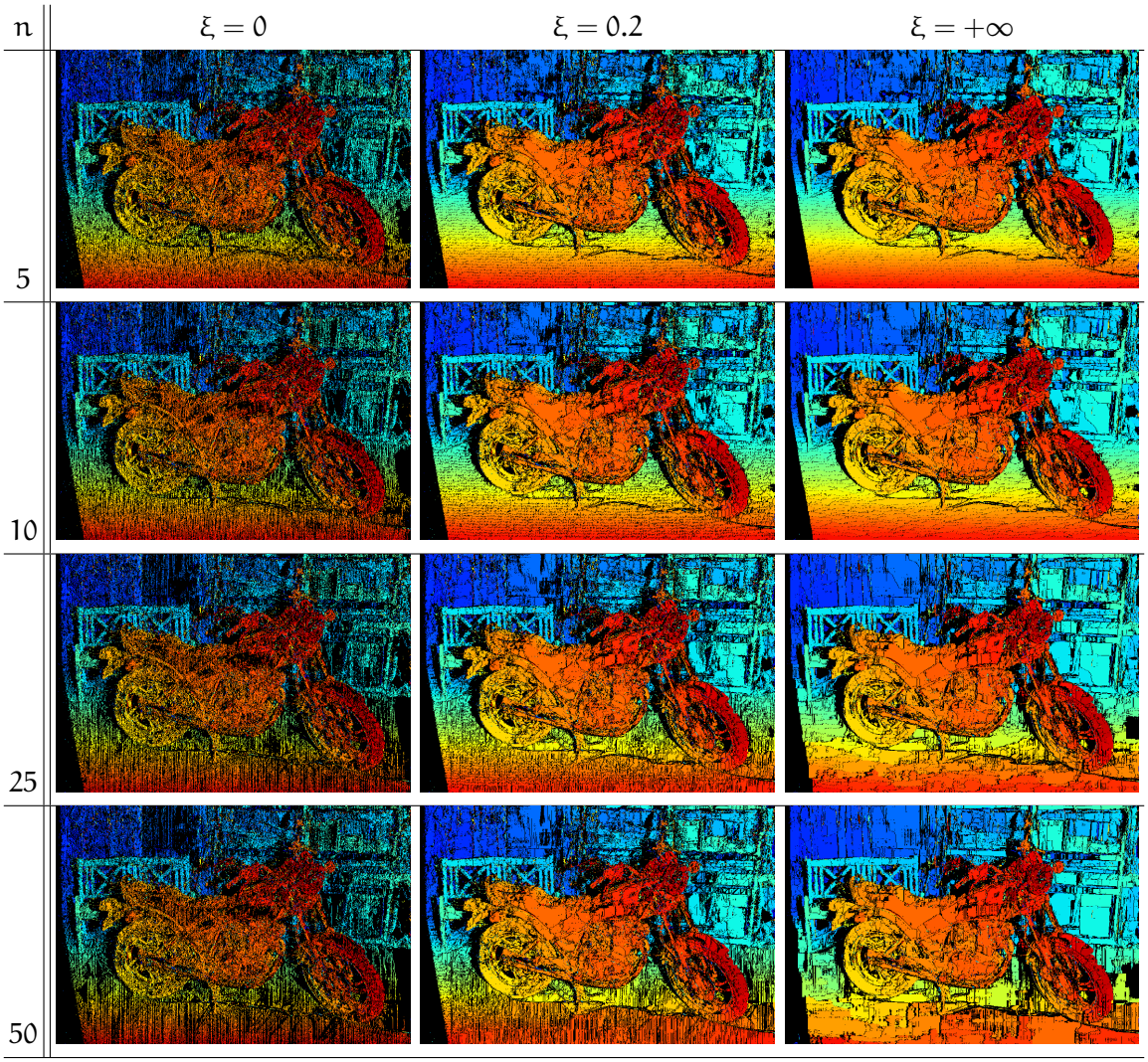


FIGURE 5.9: Initial sparse disparity maps obtained for `Adirondack` using algorithm 5.1 and varying the diffusion scope parameter n as well as the warping regularisation term ξ. Pixels shown in black are those which do not satisfy the cross checking criterion. Input image dimensions: 718×496 pixels.



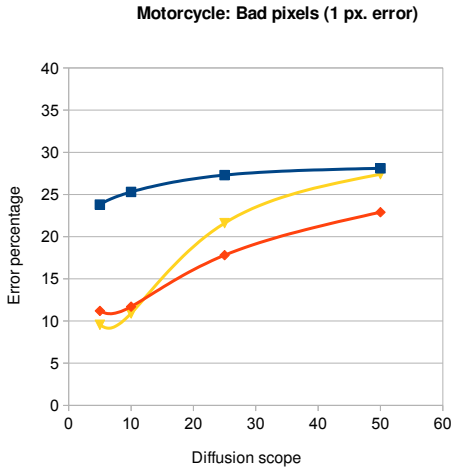


FIGURE 5.10: Initial sparse disparity maps obtained for `Motorcycle` using algorithm 5.1 and varying the diffusion scope parameter n as well as the warping regularisation term ξ. Pixels shown in black are those which do not satisfy the cross checking criterion. Input image dimensions: 741×497 pixels.

disparity measures, resulting from the method of [Scharstein et al., 2014]. As a general rule, the disparity maps increase in density as the term ξ regularising the warping paths increases. Choosing $\xi = 0$ results in the depth of the warping paths fluctuating in an uncontrolled way, which completely alters the diffused cost volume and increases the sparsity of the output disparity maps. When $\xi = +\infty$, no warping takes place, since the diffusion then operates in fronto-parallel mode. For sufficiently textured cases, like `Motorcycle`, we can obtain dense and relevant measures for a small diffusion scope n. However, we notice that employing high diffusion scopes within the fronto-parallel diffusion produces an annoying staircase effect and greatly deteriorates the accuracy of the disparity map. High diffusion scopes are necessary when dealing with images abounding with homogeneous regions, such as `Adirondack`, since otherwise there would be many ambiguities. The diffusion controlled by ξ set to 20% of the worst superimposition cost, yields disparity maps which have equivalent performances in terms of accuracy for small diffusion scopes, and significantly better performances at higher diffusion scopes, in comparison with the fronto-parallel mode. An observation which does not emerge from the quantitative results is that the errors occurring when $\xi = 0$ and $\xi = 0.2$ are mainly noisy measurements, being totally disconnected from the blocks of disparities that seem relevant.

The need for a filtering stage When interested in using sparse disparity maps to initialise the estimation process with the aim of generating complete disparity maps without holes, then the sparse disparity maps should be as accurate and as dense as possible. The following subsection presents morphological filters designed to remove erroneous disparity measures. Several problems will be addressed: one which consists of localising clusters of disparities, one which uses the occlusion reasoning to determine disparities originating from occluding contours, and one which relates the relevance of a measure to the gradient information. Finally the filtering will also play an essential role with respect to the extended diffusion mechanism presented at the end of this section...

5.2.2 Morphological filtering of sparse disparity maps

The goal of this filtering stage is the elimination of bad measures occurring in the sparse disparity maps computed according to equation 5.12. On the one hand, most of the bad measures occur as small but widely dispersed artefacts in the disparity function, especially when the warping is enabled. On the other hand, most of the good measures form smoothly evolving disparity blocks with a variable number of holes, which extend across the image plane. Furthermore, these considerably outnumber the bad measures. For that reason, we propose first, to group the disparity measures of $\mathfrak{D}^{(\text{init})}$ into pertinent clusters, and then to detect and prune the bad measures, based on the analysis of these clusters.

Clustering the disparities

Let us provide a definition for the clusters of a disparity map $\mathfrak{D}^\bullet$ which has *no* hole. Suppose that $\mathbf{p_A}$ and $\mathbf{p_B}$ are two points belonging to the same image plane. $\mathbf{p_A}$ and $\mathbf{p_B}$ belong to the same cluster if and only if there exists a path $\Gamma(\mathbf{p_A}, \mathbf{p_B})$ characterised by an ordered sequence of points $\{\mathbf{p}_k\}_{k=1}^{\ell}$ of arbitrary length ℓ such that

- The starting and ending points of the sequence are $\mathbf{p}_1 = \mathbf{p_A}$ and $\mathbf{p}_\ell = \mathbf{p_B}$.
- There is no spatial discontinuity in the path $\Gamma(\mathbf{p_A}, \mathbf{p_B})$, in that $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$ are neighbours with respect to the image plane, for any $1 \leqslant k \leqslant \ell$.
- Disparity discontinuities no larger than 1 pixel are tolerated along the path, i.e. $|\mathfrak{D}^\bullet[\mathbf{p}_k] - \mathfrak{D}^\bullet[\mathbf{p}_{k+1}]| \leqslant 1$ for all $1 \leqslant k \leqslant \ell$.

Intuition tells us, the connected components resulting from a simple threshold on the gradient of the disparity map should easily highlight most of these clusters.

Algorithm 5.2 Sparse disparity map clustering

1: **function** CLUSTERDISPARITIES($\mathfrak{D}^{(\text{init})}$, $\|\nabla \mathbf{I}_l\|$)
2: *Filling holes*
3: $\mathfrak{D}^\bullet \leftarrow \mathsf{W.T.}(\mathcal{L} := \mathfrak{D}^{(\text{init})}, \mathbf{S} := \|\nabla \mathbf{I}_l\|)$
4: *Mask holding pixels satisfying cross-checking*
5: $\mathbf{M}^{(\text{init})} \leftarrow$ Binary image highlighting all pixels set to a disparity in $\mathfrak{D}^{(\text{init})}$
6: *Masks highlighting disparity discontinuities*
7: $\mathbf{M}^{(\text{disc.E.})} \leftarrow$ Binary image highlighting all pixels satisfying $(\delta(\mathfrak{D}^\bullet) - \mathfrak{D}^\bullet) \leqslant 1$
8: $\mathbf{M}^{(\text{disc.I.})} \leftarrow$ Binary image highlighting all pixels satisfying $(\mathfrak{D}^\bullet - \varepsilon(\mathfrak{D}^\bullet)) \leqslant 1$
9: *Clusters labelling*
10: $\mathcal{C} \leftarrow \mathsf{Label}(\mathbf{M}^{(\text{disc.E.})})$
11: *Extending labelling to discontinuities*
12: $\mathcal{C} \leftarrow \mathsf{W.T.}(\mathcal{L} := \mathcal{C}, \mathbf{S} := 1 - \mathbf{M}^{(\text{disc.I.})}) \times \max\{\mathbf{M}^{(\text{disc.E.})}, \mathbf{M}^{(\text{disc.I.})}\}$
13: *Superimposing validity mask*
14: **return** $\mathcal{C} \times \mathbf{M}^{(\text{init})}$

However, we are dealing with *sparse* disparity maps. Therefore, we need a way to transform $\mathfrak{D}^{(\text{init})}$ into a full disparity map $\mathfrak{D}^\bullet$. If we interpret $\mathfrak{D}^{(\text{init})}$ as an initial image of lakes $\mathcal{L}$, it is possible to fill the holes of $\mathcal{L}$ with the disparities lying on the lakes' borders by means of the watershed transformation presented in section 3.3. The topographical surface driving the flooding is then nothing other than the gradient of the reference image $\mathbf{I}_l$ for which the disparity map is generated. Algorithm 5.2 provides the details of the disparity map clustering, with W.T. representing the watershed transformation operator and Label being the function which allocates a unique and strictly positive integer label to each connected component of the binary mask it takes as input.

The most delicate part of the procedure is the allocation of cluster labels to the image zones where disparity discontinuities occur. Algorithm 5.2 handles all single disparity discontinuities of more than 1 disparity unit at line 12, as shown by the example of figure 5.11. For the extreme

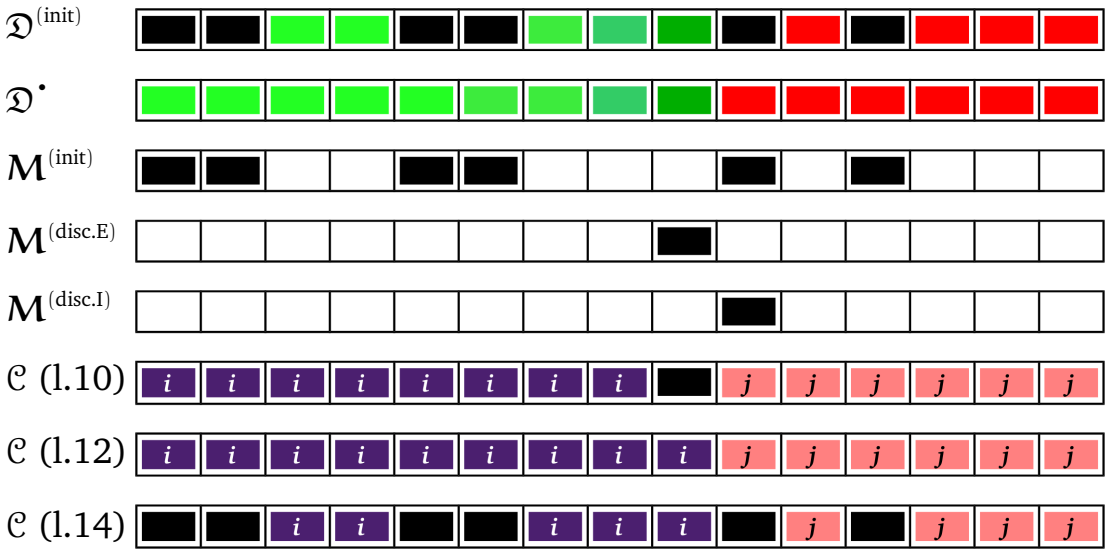


FIGURE 5.11: Illustration of the different stages of the clustering procedure implemented by algorithm 5.2 for a one-dimensional case. Pixels shown in black within the disparity map $\mathfrak{D}^{(\text{init})}$ are those which are not allocated a disparity measure.

and rare cases where such discontinuities occur consecutively, the pixels concerned receive the null cluster identifier and their disparity measures are automatically pruned from $\mathfrak{D}^{(\text{init})}$.

Pruning bad clusters

The spiky disparity measures which occur in the initial disparity map must, by construction of algorithm 5.2, belong to very small disparity clusters. At the opposite extreme, we expect the very large clusters of smoothly evolving disparities to contain only pertinent measures. For this reason, we introduce two parameters: σ_0, which represents the area below which a cluster is considered to contain bad disparity measures, and σ_1, which corresponds to the area above which the cluster is systematically considered to contain good disparity measures. We can therefore compute two binary images $\mathbf{M}_{\sigma_1}$ and $\mathbf{M}_{\sigma_0} \,|\, S(\mathbf{M}_{\sigma_1}) \subseteq S(\mathbf{M}_{\sigma_0})$ indicating respectively the localisation of large clusters to be maintained throughout this filtering stage and the localisation of those clusters which are not considered to be parasites.

In order to decide whether the clusters indicated by the binary function $\mathbf{M}_{\sigma_0} - \mathbf{M}_{\sigma_1}$ should be retained or not, it is necessary to refer to an attribute related to the input image contents. Remember from section 2.3.2 that most ambiguities arise across homogeneous regions and that the non-periodically textured regions usually constitute image zones where disparities can be computed with a high degree of confidence. Therefore, we can accept a cluster of intermediate size if and only if it spans a portion of the reference image containing sufficient gradient information. Considering that $\mathbf{C}_i$ represents the set of pixels belonging to the i-th

disparity cluster produced by algorithm 5.2, we call g_i its gradient information computed as:

$$g_i = \frac{1}{|\mathbf{C}_i|} \sum_{(x,y)\in\mathbf{C}_i} \left(\delta_n\left(\mathbf{I}_l\right) - \varepsilon_n\left(\mathbf{I}_l\right)\right)[x,y]$$

where n corresponds to the scope of the diffusion employed to generate the sparse disparity map $\mathfrak{D}^{(\text{init})}$. The reason for reusing this parameter here, is because the disparity information originating from a gradient crest is likely to propagate a maximum of n pixels in each direction of the image plane, due to the cost diffusion. Therefore, if the gradient magnitude were the only data considered when calculating the gradient information, this gradient magnitude would be likely to be allocated a fairly low value for any cluster under consideration, hence the use of the thick gradient adapted to the diffusion's scope.

To summarise, an arbitrary cluster $\mathbf{C}_i$ may be preserved at this stage of the filtering if it belongs to the binary mask $\boldsymbol{M}_{\sigma_1}$, or if it both belongs to the binary mask $\boldsymbol{M}_{\sigma_0}$ and satisfies $g_i \geqslant g_{\text{min}}$ where g_{min} constitutes the third parameter, representing the minimum gradient information requested for clusters of intermediate size. Figure 5.12 is an example which shows each step of the cluster selection.

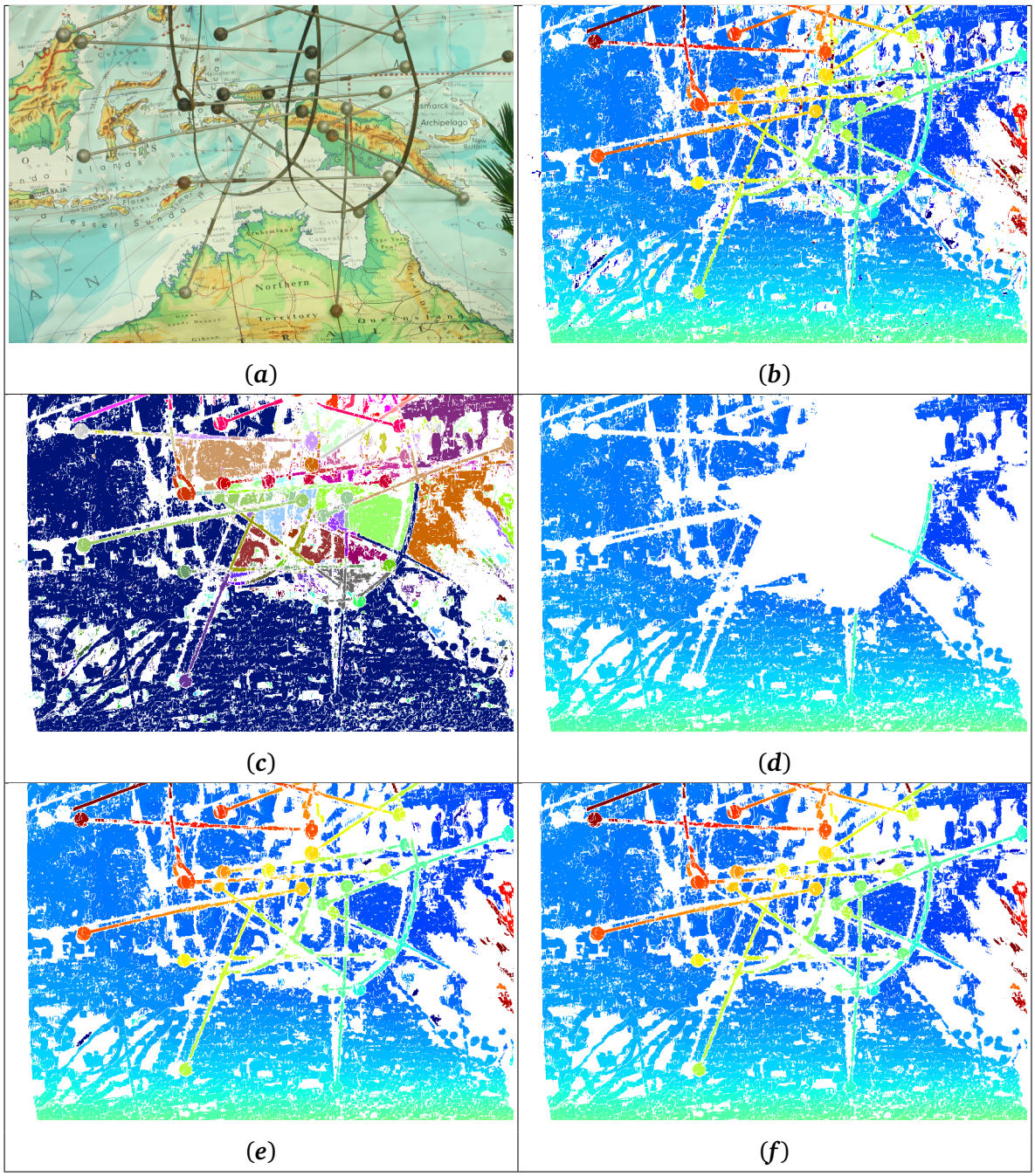


FIGURE 5.12: Pruning of bad disparity clusters for `AustraliaP`. (a) Input image (excerpt). (b) Initial disparity map $\mathfrak{D}^{(\mathrm{init})}$. ($c$) Resulting clusters map. Pixels coloured in white are excluded from the validity mask. (d) Points of $\mathfrak{D}^{(\mathrm{init})}$ belonging to $\mathbf{M}_{\sigma_1}$. (e) Points of $\mathfrak{D}^{(\mathrm{init})}$ belonging to $\mathbf{M}_{\sigma_0}$. (f) Removal of points belonging to the clusters of $\mathbf{M}_{\sigma_0} - \mathbf{M}_{\sigma_1}$ and which do not contain sufficient gradient information. **Parameters:** This initial disparity map results from the diffusion algorithm of section 5.1 with $\xi = +\infty$ and $n = 5$ for an initial image size of 1472×984 pixels. The area thresholds are set to 0.5% and 0.005% of the image area for σ_1 and σ_0 respectively. The minimum gradient information $g_{\min}$ expected for all clusters of intermediate size is set to 10% of the maximum lightness value.

Filtering out the fattening effect

Despite the use of aggregation supports constrained by the left and right image segmentations, we note that the fattening effect still surrounds the contours adjacent to very homogeneous regions. Also, this had been foreseen in section 2.1.2. Filtering based on the pruning of clusters does not suffice to handle this situation, because the pixels allocated fattened disparities are connected to those assigned a correct disparity measure.

As with the occlusion reasoning scheme presented in section 5.1.1, it is necessary to analyse the spatial distribution of the disparities on a regional basis in order to detect the abnormalities. For a diffusion scope of n pixels, those points being subject to the fattening effect lie at a distance of n pixels maximum from the region borders. Therefore, the points which are not subject to this fattening effect belong to the cells of the partition $\mathcal{L}_l$ each having undergone an erosion of size n. These cells can be represented by an eroded partition [Beucher, 2013a] which, henceforth, will be referred to as $\mathcal{L}_l^{(\varepsilon_n)}$.

We propose an algorithm which either discards or preserves the disparity clusters remaining after the pruning of bad clusters. This time though, the algorithm acts on a regional basis, rather than at global scale. Therefore, the disparity measures contained by one cluster could be discarded for one region and yet preserved for another. The decision on the preservation of an arbitrary cluster within a given region $\mathbf{R}_i$ depends on three scenarios:

1. If $\mathbf{R}_i$ is thin, then the label i should not appear in the eroded partition $\mathcal{L}_l^{(\varepsilon_n)}$. Empirical observations show that different clusters of disparities can legitimately cover such regions, which are often difficult to adequately segment. Therefore, any cluster belonging to the region is retained.
2. Otherwise, if the points having label i in $\mathcal{L}_l^{(\varepsilon_n)}$ are allocated some disparity measures, it indicates that region $\mathbf{R}_i$ contains some internal disparity information. In section 2.3.2, we established that internal disparities should be favoured over contour disparities due to the self-occlusion problem. Therefore, only the clusters to which the internal points belong are to be preserved.
3. Otherwise, region $\mathbf{R}_i$ is sufficiently large, but doesn't seem to contain any internal discriminant features useful for the measurement of disparities. Therefore, we need to resort to the contour disparities. Following our study of the gradient-based computation for the regional disparities, we opted to preserve the cluster with smallest average disparity in $\mathbf{R}_i$.

These three scenarios are illustrated in figure 5.13. Additionally, figure 5.14, which summarises the two stages of the filtering block proposed in this method, provides a good example of fattening effect.

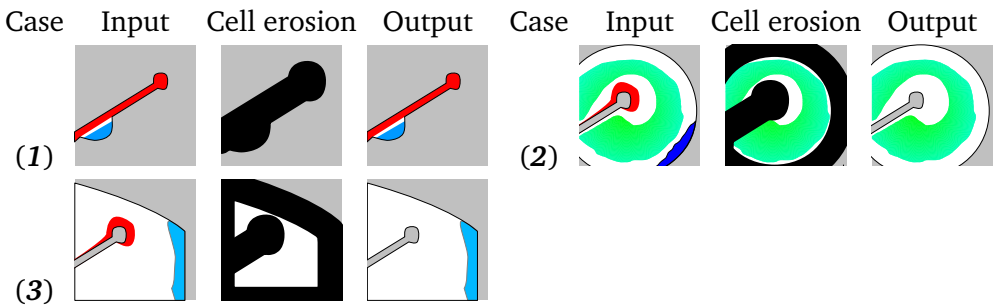


FIGURE 5.13: The logic behind the last stage of the filtering, occurring after the pruning of bad clusters. For case (1), the region of interest is extremely thin. Such regions undergo no additional filtering, and thus allow different clusters of disparities to co-exist. In case (2), the region of interest contains disparities lying far from its contours. We therefore retain only those clusters to which these disparities belong, in order to avoid both the fattening effect and self-occlusion disparities. In case (3), the disparity information stems from the contours. Therefore, we exclusively consider the cluster having the smallest average disparity in the final output.

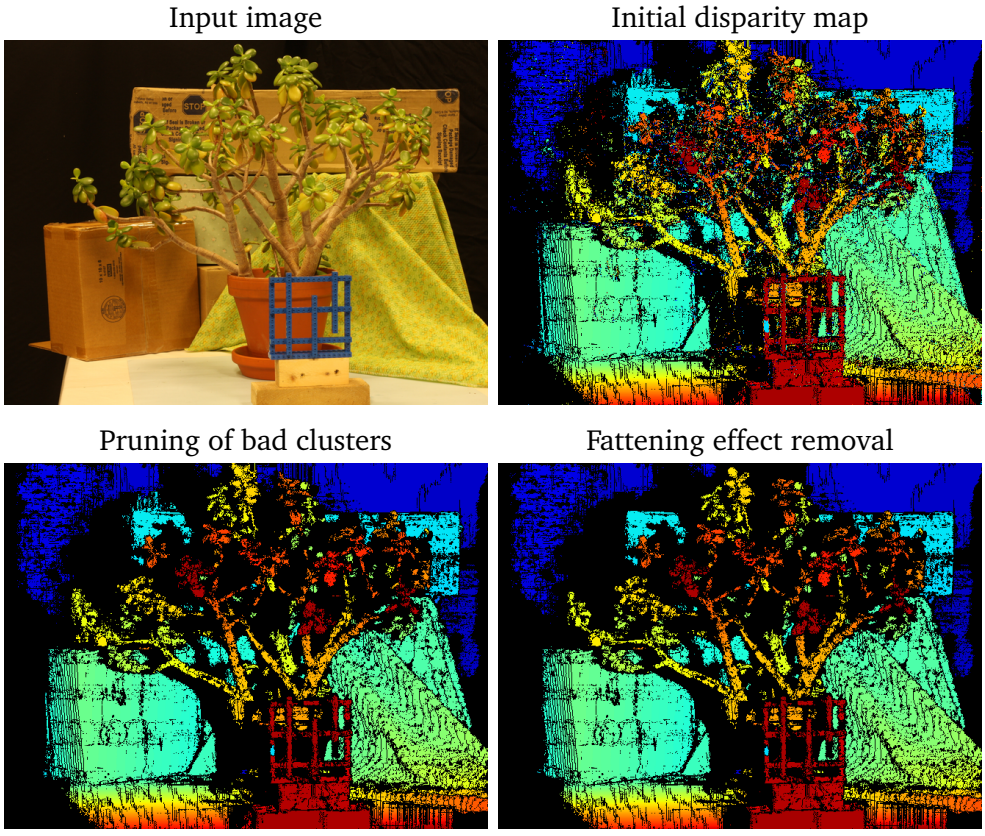


FIGURE 5.14: The filtering block proposed in this method consists of two stages: the pruning of bad disparity clusters followed by the removal of the fattening effect. The effect of the latter action is clearly visible on the `Jadeplant` test case, which contains an important number of homogeneous regions. **Parameters:** This initial disparity map results from the diffusion algorithm of section 5.1 with $\xi = 0.2$ and $n = 25$ for an initial image size of 659×497 pixels. The settings for the pruning of bad clusters are identical to those presented in figure 5.12.

Further filtering

The filtering operator proposed in this section relies on the selection of pertinent disparity clusters and on the removal of the fattening effect. The reader may have noticed that there is no constraint to ensure that the disparities will induce point matches occurring within the same pair of stereo regions. Adding this constraint to the filtering operator could improve the effectiveness of the operator, though any such constraint would assume that the ordering constraint is verified, i.e. that the points project in the same order in both stereo images, which happens decreasingly often in modern stereo databases. We also mentioned the gradient magnitude as an appropriate source of information regarding the reliability of small disparity clusters. Other useful information could have been the relevance, for a given pixel, of the cost value attached to the minimisation in equation 5.12. Of course, this cost value corresponds to a local minimum along the whole disparity axis of the DSV. But the real question is "how deep" is this minimum. Indeed, across homogeneous regions, the costs allocated to the same point but for neighbouring disparities will share very similar values. By contrast, we expect that the cost value associated with the disparity found across an irregularly textured area will be easily distinguishable from the other cost values. In that respect, the dynamic of minima [Vachier and Vincent, 1995] could play a useful role in highlighting pixels for which the superimposition cost is sufficiently discriminant.

5.2.3 Reiterated diffusions and multi-scale stereo analysis

Up to now, we have provided a cost diffusion algorithm enabling the generation of sparse disparity maps as well as a filtering operator pruning most of the erroneous measures. The examples of `Adirondack` and `Motorcycle` in figures 5.9 and 5.10 however, suggested that there is no fixed parameter systematically yielding the most dense and most accurate disparity maps. The purpose of this section is to show how the diffusion algorithm may be employed at different scales, referring here to the diffusion scope $\mathfrak{n}$, so that the resulting disparity maps are less sparse, more regularised and yet accurate.

Reiterating diffusions with increasing scopes

Small diffusion scopes employed with algorithm 5.1 yield two groups of disparity measures: those with high accuracy, especially across textured regions, and those which are wholly invalid. Fortunately, the filtering block designed in section 5.2.2 should be capable of removing the bad measures. We propose to use, in an iterative fashion, the *filtered* measures found in the sparse disparity map at a given scale, in order to constrain the diffusion of the original disparity space volume $\mathbf{D}$ at a higher scale.

Constrained diffusion Let $\tilde{\mathfrak{D}}$ be the sparse disparity map serving as input to constrain the next diffusion operation. We add the following instruction after lines 3 and 10 of algorithm 5.1, i.e.

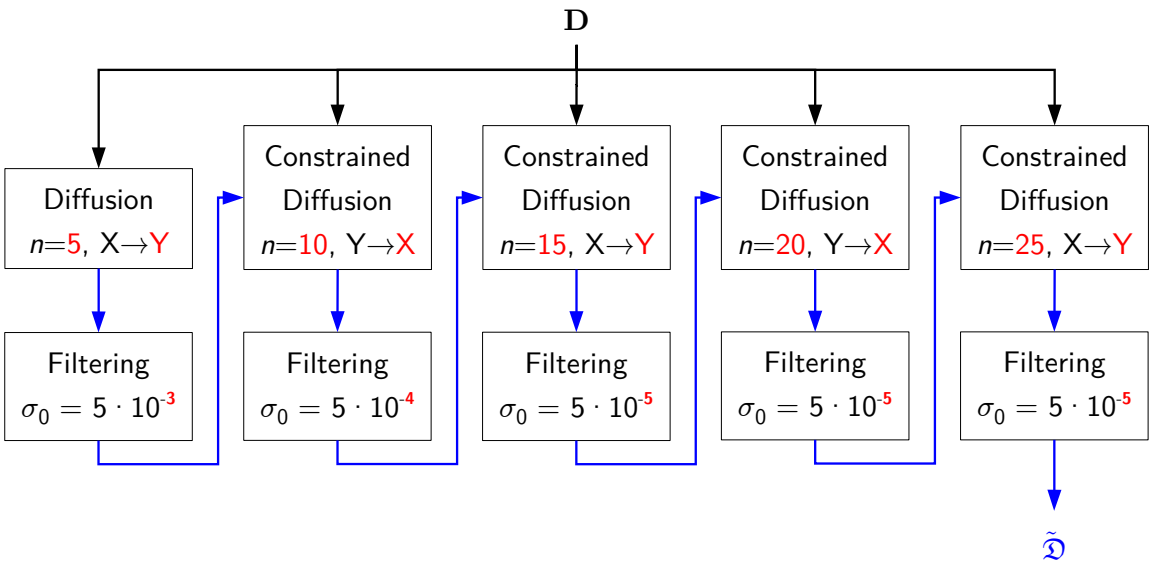


FIGURE 5.15: Reiterating diffusions with increasing scopes. Apart from the first diffusion, all diffusions are constrained by the filtered disparity map obtained at the preceding iteration, using equation 5.12 and the filtering block presented in section 5.2.2. σ_0 corresponds to the minimum area ratio expected for the small disparity clusters with respect to the entire image area. The dependence relationship $X \rightarrow Y$ indicates that the diffusion is first performed along the horizontal axis and then along the vertical axis, and vice versa for $Y \rightarrow X$.

after each update of the diffused cost volume $\mathbf{D}_{OUT}$:

$$\mathbf{D}_{OUT}[x, y, \tilde{\mathfrak{D}}[x, y]] \leftarrow 0, \text{for all } (x, y) \text{ having a measure in } \tilde{\mathfrak{D}} \tag{5.13}$$

In other words, the disparity measures found in $\tilde{\mathfrak{D}}$ now systematically induce a trace in $\mathbf{D}_{OUT}$ along which the superimposition costs are set to zero. The voxels which are part of this superimposition trace are never updated, which means that:

- the disparities allocated to the pixels onto which they project remain unchanged, once the diffusion process is complete.
- at each iteration of the directional diffusion, a null cost is diffused to their neighbour voxels. Therefore, the pixels onto which these neighbour voxels project should receive disparities close to those found in their neighbourhood, after the diffusion terminates.

The full multi-scale diffusion pipeline is schematised in figure 5.15. We tested five different scales with a diffusion scope ranging from $n = 5$ pixels to $n = 25$ pixels. Remember that figures 5.9 and 5.10 showed that higher scopes do not seem to add further relevant information. We impose a more significant filtering for the smaller diffusion scopes, so that only the biggest disparity clusters are preserved. This is in order to maximise the chances of pruning all the bad measures.

Figure 5.16 shows the result of this multi-scale diffusion on `Adirondack` and `Motorcycle`.

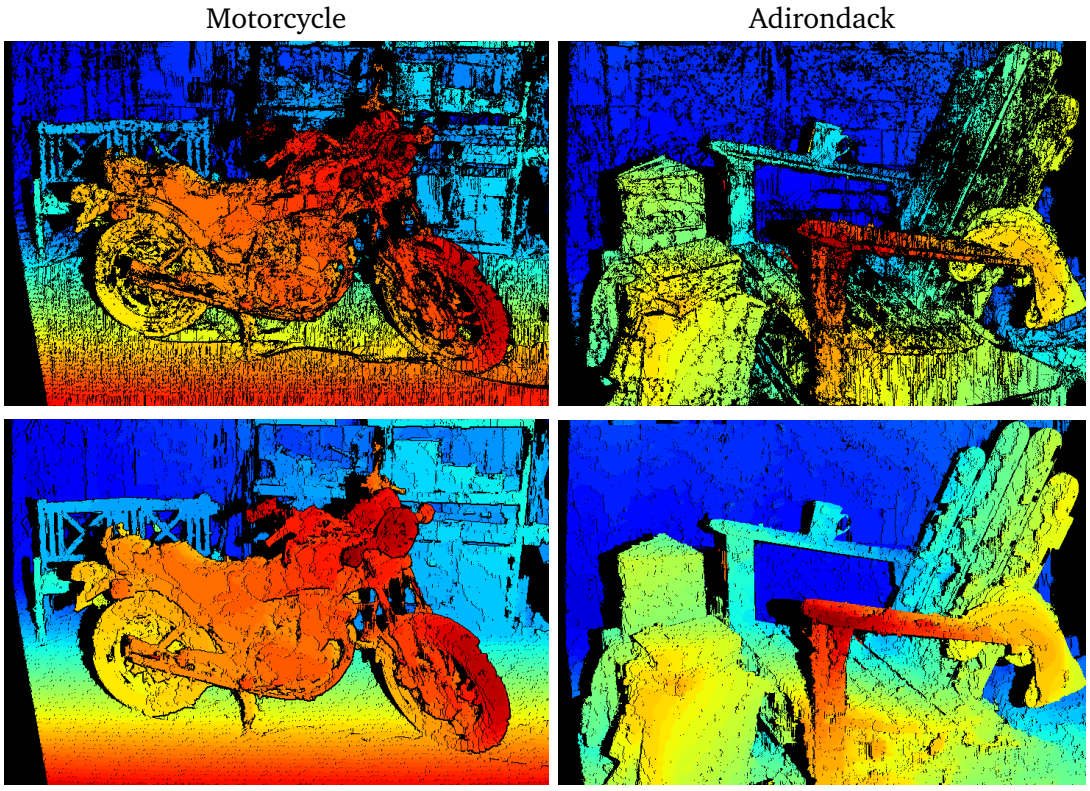


FIGURE 5.16: Top row: Filtered disparity maps resulting from diffusion algorithm 5.1 with $n = 25$ and $\xi = 0.2$. Bottom row: Disparity maps resulting from the chaining scheme represented in figure 5.15, using $\xi = 0.2$.

Observations The first noticeable change between the single diffusion at scope $n = 25$ and the chaining of these reiterated diffusions is of course the strong reduction of sparsity which we expected. For `Motorcycle`, the disparity map seems impressive since details almost indiscernible in the original image, such as the side-view mirror, are clearly accentuated. However, the small diffusion scope already provided very good measurements. If we now look at the `Adirondack` example, we can see some bumps appearing across the homogeneous regions. These bumps are due to initial measurements at fine diffusion scopes, which initially seemed correct, but which proved inaccurate after completion of the multi-scale densification. Therefore, the disparity maps produced could still lack regularity. Smoothness can be achieved using global minimisation frameworks driven by equation 2.8, but this is computationally very expensive. Research in optical flow [Fleet and Weiss, 2006] established that coarse-to-fine refinements approaches, being the exact inverse of what we have just proposed, typically help in solving the ambiguities that arise from taking measurements at fine scales. In the next paragraph, we investigate how coarse-to-fine refinement can be performed using our diffusion algorithm. Note, however, that the fine-to-coarse iteration technique we have just presented will produce the best results if used *after* the coarse-to-fine refinement.

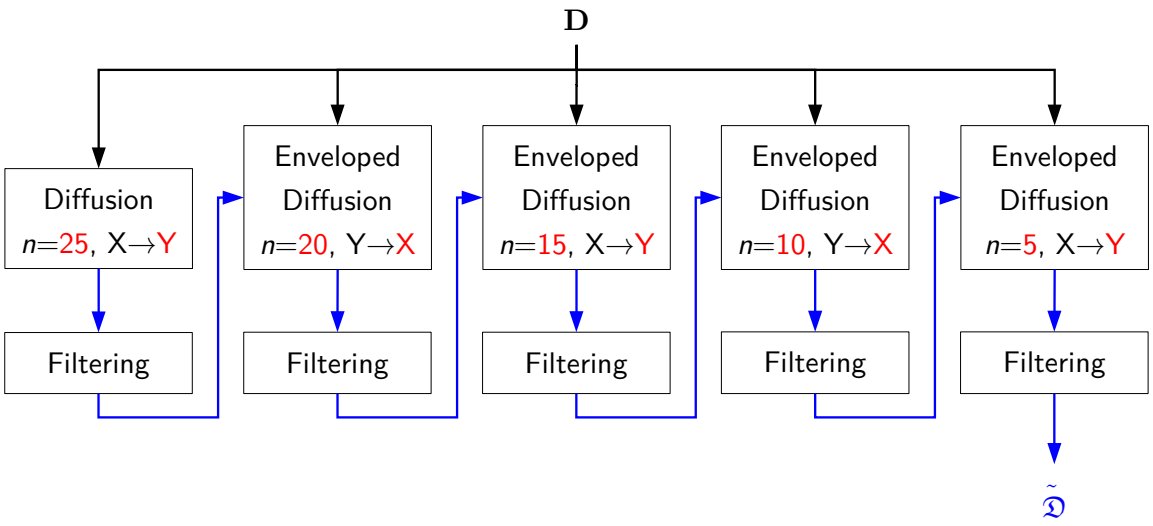


FIGURE 5.17: The coarse-to-fine refinement system proposed using our diffusion algorithm is the reverse of the multi-scale diffusion approach using increasing diffusion scopes. Note that the enveloped diffusion operator enforces that the measures available in the input disparity map, are allowed a variation of a few pixels in the output disparity map.

Coarse-to-fine refinement

The coarse-to-fine refinement pipeline based on our diffusion algorithm is illustrated in figure 5.17. Contrary to the fine-to-coarse procedure elaborated previously, it is no longer sensible to impose that the disparities found at a higher scale of the diffusion be *exactly* the same at a finer scale. This is because the disparities measured at the coarser scale, are more approximate than those obtained at finer scales. However, the disparities measured at high diffusion scopes never differ too much from the actual disparities. Therefore, the dependence between two consecutive diffusions should be modelled such that the disparity measures obtained after the i-th iteration, for a diffusion scope n_i, define an *envelope* inside which we expect to find the disparities after the subsequent iteration $i+1$, for a diffusion scope $n_{i+1} < n_i$. Furthermore, this envelope should be related to the variability of the measure obtained at iteration i, which should gradually decrease to zero as the scale diminishes.

In order to take this envelope into account within the diffusion operator at iteration $i+1$, lines 3 and 10 of algorithm 5.1 must be followed by the this next instruction:

$$\mathbf{D}_{\mathrm{OUT}}[x,y,d] \leftarrow \begin{cases} c_{\max} & \text{if } d' = \tilde{\mathfrak{D}}[x,y] \text{ and } |d-d'| > \Delta(n_i) \\ \mathbf{D}_{\mathrm{OUT}}[x,y,d] & \text{otherwise} \end{cases} \tag{5.14}$$

where $c_{\max}$ represents the worst superimposition cost possible for the DSV and $\Delta(n_i)$ indicates the tolerated variation of disparity from the disparity measures performed at iteration i. When

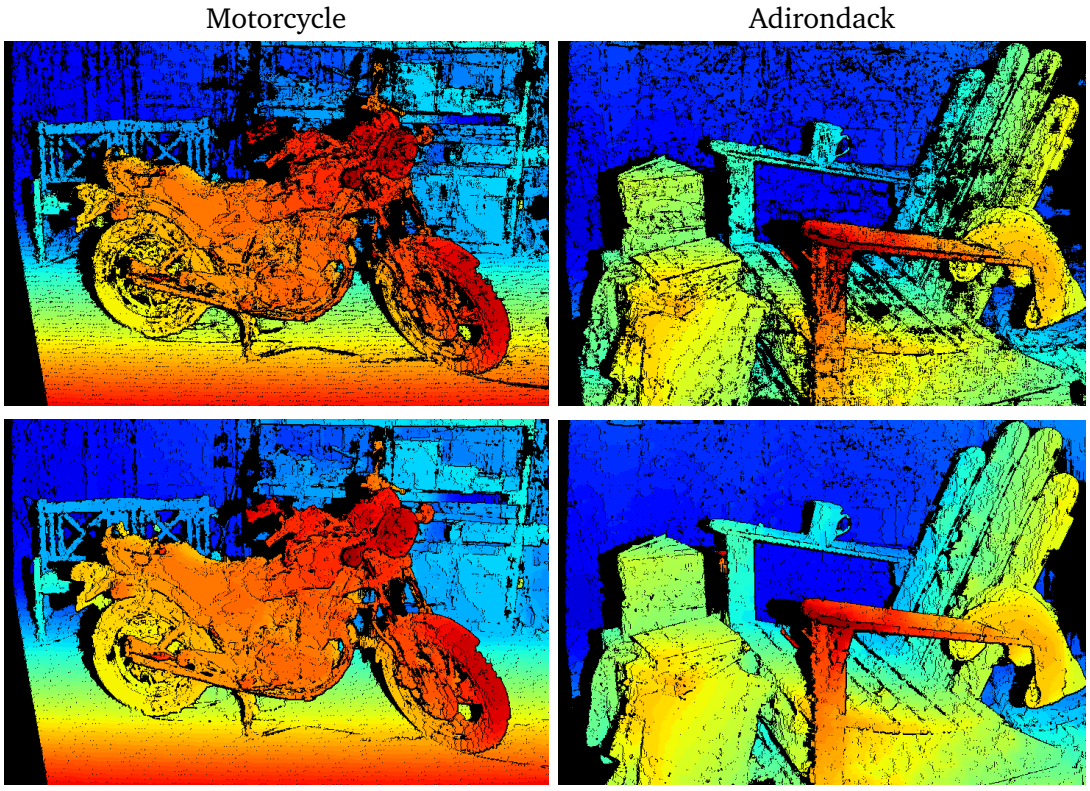


FIGURE 5.18: Top row: Filtered disparity maps resulting from the chaining scheme represented in figure 5.17, using $\xi = 0.2$. Bottom row: Disparity maps resulting from the chaining of coarse-to-fine refinement and iterative diffusions of increasing scopes.

employing a diffusion scope of $n = 25$ pixels across the Middlebury 2014 dataset, we found that the average error rate for the worst 10% of bad measures *after* the filtering stage was 2.55 pixels (quarter resolution), which confirms that the bias between the measured and actual disparities is a matter of a few pixels. In our experiments, we chose to tolerate slightly more variation than the aforementioned mean error and therefore, we opted for $\Delta(n_i) = \lfloor n_i/5 \rfloor$.

Observations The top row of figure 5.18 shows the result of the coarse-to-fine refinement on the `Adirondack` and `Motorcycle` test cases. The images are slightly sparser than those obtained using the fine-to-coarse scheme, but it can be observed that the disparities related to the ground region have been corrected for `Motorcycle` and that the disparity function across the armrests and the pedestal of the chair has been considerably smoothed. The bottom row of figure 5.18 shows the output of the fine-to-coarse procedure starting directly from the constrained diffusion of size $n = 10$, with the disparity maps presented in the top row as an initial constraint. A further example is provided for the `Recycle` test case in figure 5.19 showing the intermediate disparity maps after each step of the refinement. The benefits of the coarse-to-fine refinement are clearly noticeable when comparing the disparity map ensuing from the chaining of both multi-scale systems and that resulting from the fine-to-coarse expansion only.

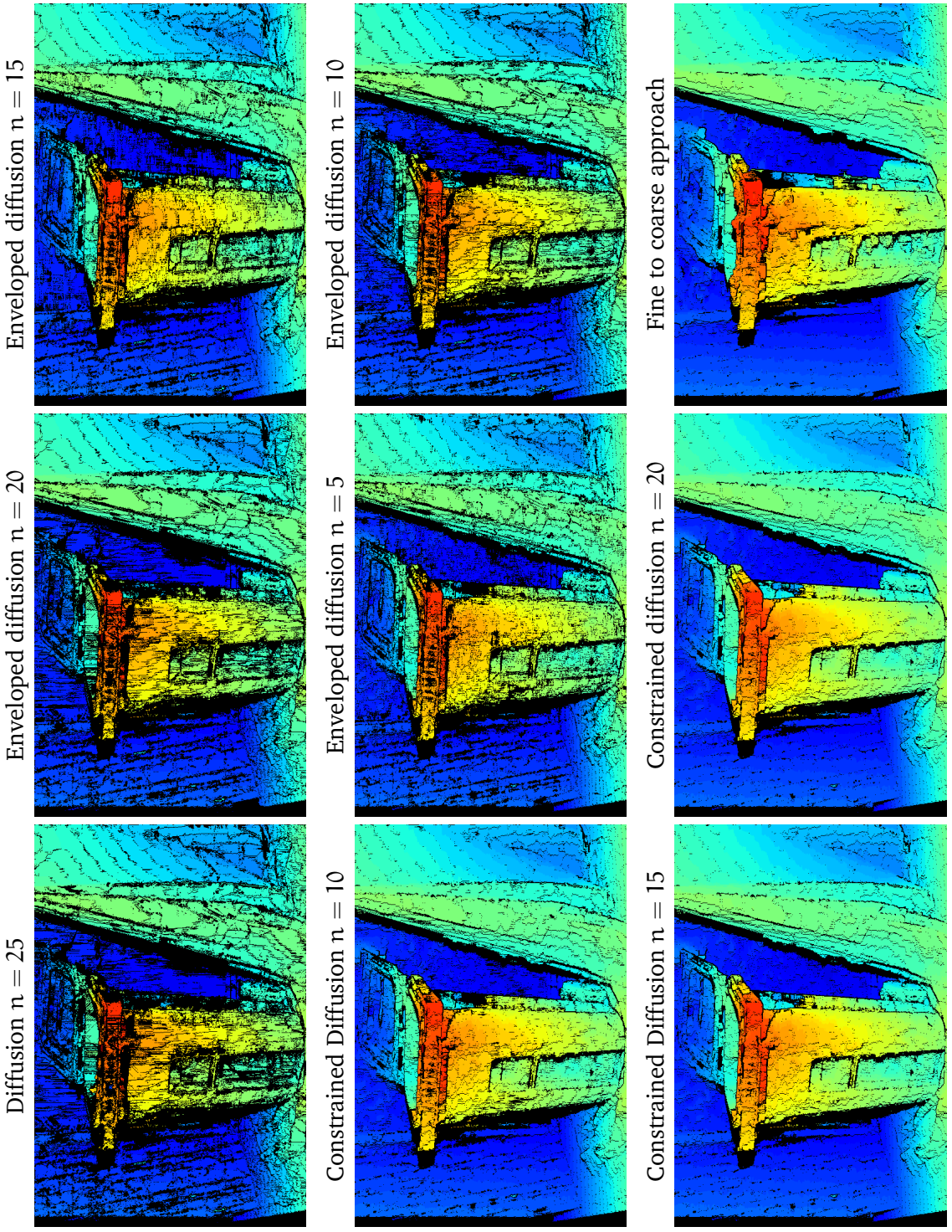


FIGURE 5.19: Visualisation of intermediate disparity maps for the full multi-scale generation of disparity map for `Recycle`.

Coarse-to-fine refinement, followed by fine-to-coarse densification

To summarise, we have devised two complementary multi-scale pipelines. It is first necessary to employ the coarse-to-fine refinement system displayed in figure 5.17 in order to regularise the disparity maps and to prevent the occurrence of artefacts arising from the measurements at fine scale. Then, the fine-to-coarse densification system shown in figure 5.15 has to be used to reduce the sparsity of the disparity map while taking account of the image superimposition costs.

Quantitative evaluation We refer the reader to section A.3 of the appendix, which provides, in terms of accuracy, an analysis of the evolution of the disparity maps through the different stages of this multi-scale computation.

Summary

The issue which makes the measurement of disparities across a pair of stereo images a challenging task, is the abundance of homogeneous areas and occlusions. In this study, we proposed two alternative methods which exploit the segmentations of the left and right stereo images, so as to properly handle these two phenomena. Common to both alternatives is the constraint imposed on the correlation supports, which prevents costs from different regions mixing with each other, thus hindering the integration of the costs of occluded image areas with those of correctly superimposed image areas. Another important result of this work was the manipulation of occluding contour disparities which caused incorrect regional disparities, or incorrectly propagated contour disparities in both homogeneous and occluded areas when performing the cost diffusion.

The results produced by the two algorithms are distinct. The regional disparities are used to generate regional disparity maps, which are complete disparity functions; smooth because they are flat across each region, sharp at the segmentation boundaries, and perceptually appealing if the geometrical assumptions concerning the scene (objects relatively fronto-parallel to the camera), or the acquisition setup (low-baseline) are satisfied. Furthermore, they are comparatively easy to compute. However regional disparities are not sufficiently meaningful when the objects in the scene are very tilted with respect to the image plane. In that case, it is preferable to deduce disparities on a point basis from the a diffused disparity space volume. We observed the effect of the different parameters and the accompanying filtering operator on the resulting disparity maps. By using one single diffusion, it is possible to obtain very accurate disparity measures, despite some lack of information across homogeneous regions. Re-iterating diffusions helps with reducing the sparsity of the disparity measures across homogeneous regions and with allocating a disparity value to virtually all image points, except those lying in the occluded image areas. At the end of this summary, we provide a comparative chart to facilitate weighing the features of both methods.

The following chapters explain how disparity measures can be used to compute the equivalent segmentations of stereo images and how they can be integrated within the estimation of the final depth maps.

	Regional disparities	Regional aggregations
Correlation supports		
Investigated cost measures	Mean of absolute differences (luminance or gradient-based).	Hamming distance between Census-transformed images.
Stereo matching strategy	Pixels belonging to the same region are matched in one block and thus receive the same disparity.	Each pixel is matched separately.
DSV Aggregation supports	1. span only one disparity plane (fronto-parallel assumption). 2. are the same for any voxel lying in the same disparity plane and regional intersection.	1. span several disparity planes. 2. may be different for each voxel of the DSV. 3. are constrained by the regional intersections and maximum diffusion scopes.
Occlusions processing		
Costs in occluded area	do not aggregate with others due to correlation supports constrained by the boundaries provided by the left and right segmentations.	
Fattening of occluding contour disparities	is handled in the gradient-based computation by virtue of contour-based occlusion reasoning.	occurs only in homogeneous areas and is detected and removed during the filtering stage.
Occlusion area detection	deduced from the regional disparity maps, at a coarse segmentation level.	performed via cross-checking.
Measure(s) assigned to points lying in occluded areas	Regional disparity of the region to which the points belong.	None.
Homogeneous areas processing		
Measure(s) assigned to points lying in homogeneous areas	Regional disparity of the region to which the points belong.	None if cross-checking is not satisfied. Ambiguities can be overcome using the coarse-to-fine-to-coarse multi-scale diffusion system.
Working conditions		
Geometrical assumptions	1. Regions are (almost) fronto-parallel to the camera. 2. A low baseline has been used to acquire the stereo images. 3. Intra-region ordering constraint applies.	None.
Parameters	Minimum overlap ratio required between matched regions.	1. Diffusion maximal scope. 2. Warping regularisation term. 3. Filtering area and gradient information terms. 4. Multi-scale envelope variation term.
Computational load		
Memory load	Small. Only a few 2D images have to be kept in memory (suitable for HD processing).	Significant. Full disparity space volume required to be stored in memory.
Time load (Python 2.7 environment, 1x Core i5 at 2.7 GHz)	Relatively small: less than 60 sec. to process a full HD stereo pair.	Depends on the diffusion scope. Around 300 sec. for a diffusion of $n = 25$ pixels on an SD stereo pair.
End-results quality		
Perception	1. Complete disparity maps. 2. Very sharp boundaries. 3. Consistent regional depth ordering.	1. Sparse disparity maps. 2. Only occluded areas are left without a measure after multi-scale expansion.
Accuracy	1. Optimal in fronto-parallel scenarios. 2. Approximate for regions being slightly tilted. May be refined using finer segmentation. 3. Bad for highly tilted regions. Main limitation: no information available about the bias induced by regional disparities.	1. High precision for disparity maps obtained using a single diffusion. 2. Still good precision after the multi-scale expansion. 3. Very low RMS errors.

Résumé du chapitre 6

Ce sixième chapitre est consacré à l'utilisation des mesures de disparité dans le cadre de la génération de co-segmentations : étant donné la segmentation associée à la vue de référence, le but du problème est de trouver la segmentation équivalente associée à l'autre vue de la paire stéréo. Pour ce faire, chaque région issue de la partition de l'image de référence est associée à un marqueur. Ce marqueur est ensuite décalé en fonction des mesures de disparité disponibles, de telle sorte à être inclus dans la future région correspondante de l'image cible. Afin d'identifier les régions de l'image de référence dont proviennent ces marqueurs, nous préservons leur label durant le transfert. Enfin, un mécanisme supplémentaire garantit que les régions de l'image cible, occultées dans l'image de référence, puissent être attribuées à de nouveaux marqueurs.

Nos expériences montrent que les disparités régionales, utilisées comme mesures de disparité, fournissent de très bons résultats, ces derniers pouvant servir à l'extraction de disparités ponctuelles, le long des contours de régions. Dans le cas des cartes de disparités sporadiques, les résultats sont globalement bons, mais peuvent être sujets à quelques artefacts, dont nous expliquons les causes dans ce chapitre. Quelques pistes quant à leur résorption sont par ailleurs suggérées.

Chapter 6

Equivalent Stereo Partitions

Until now, the left and right segmentations of a pair of stereo images have been generated using identical parameters, though independently of each other. Equivalent partitions, which could be beneficial either to the filtering and multi-scale cost diffusion discussed in section 5.2, or to the final estimation phase, are indispensable if regional correspondences are to be used to further prevent the establishment of bad point correspondences.

This chapter introduces the co-segmentation of stereo images, which consists in producing equivalent stereo partitions. In equivalent partitions, corresponding objects are segmented in exactly the same way, and are allocated the same label. One type of co-segmentation is the binary co-segmentation, used to extract the same object from a collection of images acquired in different environments. Several methods exist, including that of [Joulin et al., 2010], who classify the cells of an image over-segmentation into two categories; "object" or "background" by use of a discriminative clustering technique. [Rubio et al., 2012] label the pixels of a collection of images in such a way that each label represents a unique type of object. The labelling remains consistent with respect to the regions to which a pixel belongs, depending on the segmentation levels under consideration. This labelling also induces consistent region matches across the image collection, so that the colour and internal SIFT descriptors of the matched regions concur. This last method also analyses the topology of the segmentation graph to compare the region nodes, as was also the case in [Gomila, 2001], where an approach to perform equivalent multi-object segmentations across video sequences is proposed. With reference to stereo images, [Bleyer et al., 2011] propose an algorithm capable of simultaneously estimating depth maps and generating equivalent segmentations between stereo images, but it is computationally expensive and requires generation of the full depth map to obtain equivalent segmentations. An approach based on the watershed transformation was proposed earlier: it required no initial disparity measure, and consisted of transferring the segmentation markers along with their labels from one image of the stereo pair to the other [Beucher, 1990]. Our proposed method of generating equivalent stereo partitions expands on this work, adapting it to conform to modern stereo databases.

6.1 Algorithms

Let $\mathcal{L}_l$ be the partition of the reference image $\mathbf{I}_l$, for which the disparity map is denoted by $\mathfrak{D}$. We call $\mathcal{L}_l[x,y]$ and $\mathfrak{D}[x,y]$ the regional label and disparity allocated to pixel $\mathbf{p} = (x,y)$, respectively. Assuming that all pixels of $\mathfrak{D}$ are assigned correct disparity measures, we can determine their locations in the equivalent partition of the stereo pair $\tilde{\mathcal{L}_r}$.

Algorithm 6.1 Partition transfer and occlusion map generation

```
1:  function TRANSFERPARTITION(L_l, D)
2:      Initialise right image partition
3:      L̃_r ← Label map initialised to the size of L_l, all labels set to 0.
4:      Initialise right image disparity map
5:      D̃_r ← Disparity map initialised to the size of D, all disparities set to 0.
6:      Initialise occlusion mask for the left image
7:      M ← Binary mask initialised to the size of L_l, all pixels deactivated.
8:      Browse all pixels
9:      for all (x, y) belonging to the image domain do
10:         d' ← D[x, y]
11:         if x − d' < 0 then
12:             M[x, y] ← 1                                  ▷ Border occlusion
13:         else
14:             d ← D̃_r[x − d', y]
15:             if d' > d then
16:                 L̃_r[x − d', y] ← L_l[x, y]               ▷ Label transfer
17:                 D̃_r[x − d', y] ← d'
18:                 if d > 0 then
19:                     M[x − d' + d, y] ← 1                 ▷ Object occlusion
20:             else
21:                 M[x, y] ← 1                              ▷ Object occlusion
22:     return L̃_r and M
```

6.1.1 Partition transfer from a disparity map

Initially, we set all pixels in $\tilde{\mathcal{L}_r}$ to zero, i.e. without an attributed region label. The disparity of $\mathbf{p}$ is $d = \mathfrak{D}[x,y]$, therefore we expect that $\mathbf{p}$ projects with label $\mathcal{L}_l[x,y]$ in $\tilde{\mathcal{L}_r}[x-d,y]$. However, this is only possible if $\mathbf{p}$ is not occluded by another point in the right image of the stereo pair $\mathbf{I}_r$. In order to establish whether or not this is the case, it suffices to recall from equation 5.7 that $\mathbf{p}$ is occluded by another point if and only if there exists a pixel $\mathbf{p}' = (x',y)$ with $x' > x$ and projecting onto $(x-d,y)$ in $\tilde{\mathcal{L}_r}$. In the affirmative, if $d' = \mathfrak{D}[x',y]$, we have an equality between $x-d$ and $x'-d'$, which implies that $d' > d$, i.e. that, in the rectified stereo configuration, $\mathbf{p}'$ is closer to the camera than $\mathbf{p}$. As a result, equation 6.1 summarises the transfer of the left image

partition to the right image partition:

$$\begin{aligned} d'[x,y] &= \max_{d} \{\mathfrak{D}[x+d,y] \times \bar{T}(\mathfrak{D}[x+d,y]-d)\} \\ \tilde{\mathcal{L}_r}[x,y] &= \mathcal{L}_l[x+d'[x,y],y] \end{aligned} \tag{6.1}$$

where $\bar{T}$ represents a binary indicator function, such that $\bar{T}[x] = 1 \Leftrightarrow x = 0$, and $d'[x,y]$ is the disparity of the point projected onto (x,y) in the right image of the stereo pair, $\mathbf{I}_r$. In fact, both the occlusion map and label transfer can be computed in a single pass, by dealing only once with each pixel of the image plane, using algorithm 6.1.

We notice that the pixels of $\mathcal{L}_l$ which are occluded in the right view of the stereo pair cannot propagate their labels to the transferred partition $\tilde{\mathcal{L}_r}$, which is a desirable property. However, pixels in the right view which are occluded in the left view, will also receive no label at all. This is problematic if a full region of the right image is occluded in the left image, since the watershed transformation generated from the incomplete label map $\tilde{\mathcal{L}_r}$ would merge these regions with non-occluded regions. The purpose of the next algorithm is to prevent such errors. Finally, we shall explain how disparity measures resulting from the methods presented in chapter 5 can be employed in conjunction with the proposed transfer mechanism.

6.1.2 Handling occluded regions

The regions which appear *only* in the right view of the stereo pair, therefore excluding corresponding regions, which appear in the right view and partly in the left view, are part of the independent partition $\mathcal{L}_r$ computed for $\mathbf{I}_r$. We explained that when transferring the labels of $\mathcal{L}_l$ to $\tilde{\mathcal{L}_r}$ using disparity map $\mathfrak{D}$, such regions could not receive a label or, in other words, could not be *marked*.

Let $\mathcal{L}_E$ be the initial label map used to generate the watershed segmentation of $\mathbf{I}_r$, equivalent to segmentation $\mathcal{L}_l$. We expect $\mathcal{L}_E$ to be of the form:

$$\mathcal{L}_E = \text{“Markers from the left view”} + \text{“Markers for occluded right view objects”}$$

The markers from the left view correspond directly to $\tilde{\mathcal{L}_r}$ and are already allocated the desired labels. Concerning the second term of this addition, a cell of partition $\mathcal{L}_r$ allocated an arbitrary label ℓ, denotes a right view object being totally occluded in the left view, if for all $(x,y)|\mathcal{L}_r[x,y] = \ell$, the relation $\tilde{\mathcal{L}_r}[x,y] = 0$ holds. Such cells may therefore be extracted as the residue of the cell-reconstruction [Beucher, 2013a] of $\mathcal{L}_r$ by the partition $\tilde{\mathcal{L}_r}$. After their extraction, it is necessary to allocate them new labels, distinct from those included in $\tilde{\mathcal{L}_r}$. The label map resulting from this process then only needs to be appended to $\tilde{\mathcal{L}_r}$ to form the initial image of markers $\mathcal{L}_E$ driving the equivalent watershed segmentation of the right image $\mathbf{I}_r$. Figure 6.1 summarises the partition

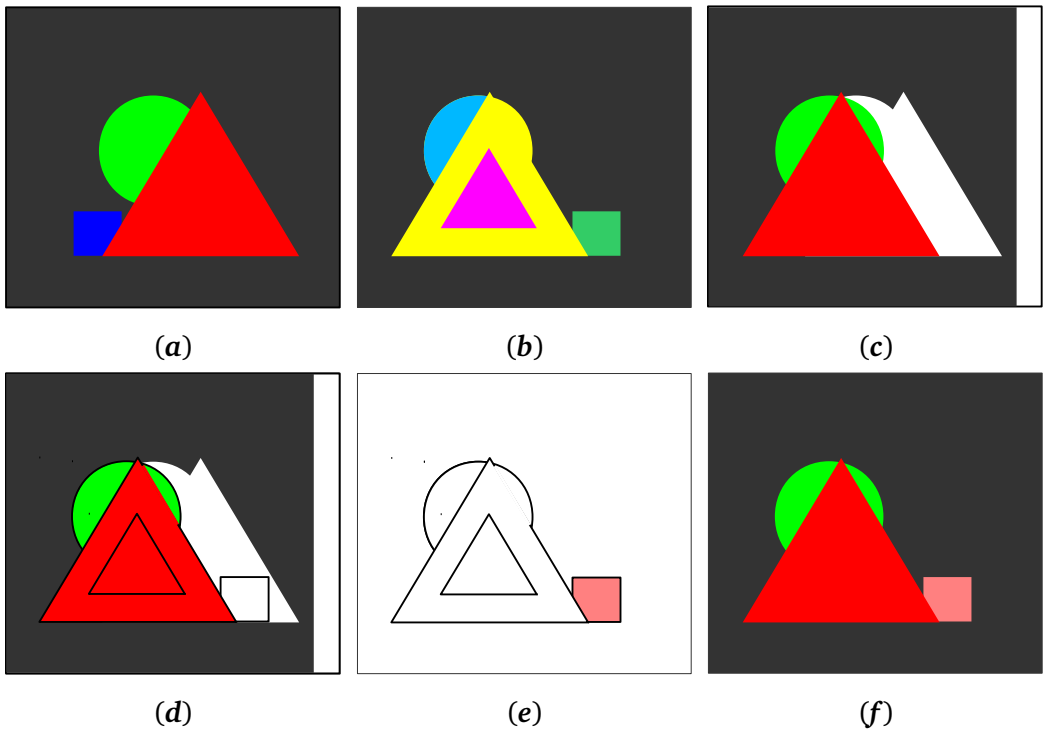
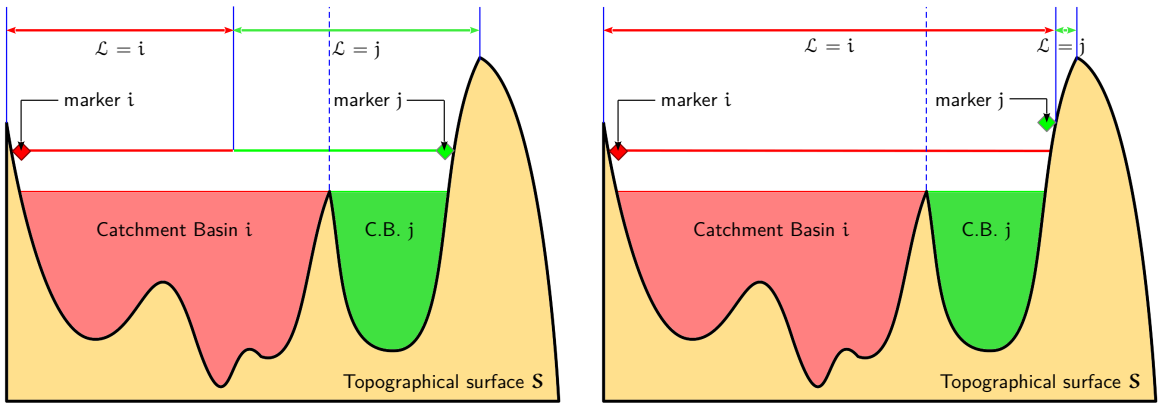


FIGURE 6.1: Illustration of the partition transfer from the left to the right view of the stereo pair, and occlusion handling. (*a*) and (*b*) represent the initial left and right partitions of the stereo pair, $\mathcal{L}_\mathrm{l}$ and $\mathcal{L}_\mathrm{r}$ respectively. (*c*) $\mathcal{L}_\mathrm{l}$ is transferred to the right view according to a disparity map $\mathfrak{D}$ to form $\tilde{\mathcal{L}_\mathrm{r}}$. Some pixels of $\tilde{\mathcal{L}_\mathrm{r}}$ are not allocated any label. These are displayed in white. Objects in the left view occluded in the right view immediately disappear from $\tilde{\mathcal{L}_\mathrm{r}}$. (*d*) There are regions of $\mathcal{L}_\mathrm{r}$ where no pixel has a label. (*e*) These regions are allocated new labels, distinct from those contained in $\tilde{\mathcal{L}_\mathrm{r}}$ to form $\mathcal{L}_\mathrm{O}$. (*f*) The watershed transformation of the right image gradient driven by the initial markers $\mathcal{L}_\mathrm{E} = \max(\tilde{\mathcal{L}_\mathrm{r}}, \mathcal{L}_\mathrm{O})$ yields an equivalent segmentation of $\mathcal{L}_\mathrm{l}$ for the right view.

transfer from the left to the right view of the stereo pair, and the handling of occluded regions.

6.1.3 Using available disparity measures

So far, we have defined the main principles of our co-segmentation algorithm: transferring available regions from one view to the other and treating occlusion issues with care. However, with respect to our method of computing disparity maps, it is essential to produce such co-segmentations *without* the need to resort to a complete or accurate disparity map. As mentioned in the introduction, the original co-segmentation method devised by [Beucher, 1990] does not require any disparity information at all. The markers responsible for generating the segmentation of the reference image originate from the gradient's minima. They are then superimposed on the gradient of the second image. Afterwards, a gradient descent is used to make these markers trickle down the slopes of the second image gradient, until they each reach a local minimum. This last point is essential in order to avoid segmentation artefacts brought about by badly placed



(a) (b)

FIGURE 6.2: Badly placed markers can be the source of segmentation errors. There are two possible scenarios: The competing markers lie (a) at the same altitude or (b) at different altitudes with respect to the topographical surface S. But in both cases, the markers are outside the catchment basins they intended to segment. Since their altitude is higher than the crest representing the watershed which separates catchment basins i and j, the watershed line induced by the two markers does not necessarily belong to the watershed generated from the minima of S. This watershed line is obtained by their SKIZ within the mask described by $\{(x, y) \mid S[x, y] < h\}$ for the altitude being considered, h.

markers, as demonstrated by figure 6.2.

This method has been employed on stereo images with a small baseline, and mainly composed of wide regions. As a result, each marker involved in the segmentation of the left view, could be transferred to the right view without altering its position, since it would already be placed over its target catchment basin. The reason we need prior disparity data is precisely because the aforementioned assumptions no longer hold true. Nonetheless, it is worth remembering, as explained in this study, that it is not necessary to employ markers which each cover the full region they are required to describe, and that, in order to avoid false segmentation lines, the transferred labels should always be projected onto the minima of the topographical surface controlling the watershed segmentation of the right view.

From regional disparities to co-segmentation

Let us explain how to derive co-segmentations from the regional disparities presented in section 5.1. We recall that regional disparities are approximate. Therefore, if we transfer the partition of the left view according to its regional disparity map, some pixels in $\tilde{\mathcal{L}}_r$ are likely to be allocated wrong labels, especially if they originate from cell borders. In order to reduce the risk of bad labellings, the transfer can be limited to the pixels lying at a sufficient distance from the cell borders. To proceed, one can use the adaptive erosion introduced in section 4.2.1 and apply it to each cell of the transferred partition. The co-segmentation procedure based on the regional disparities is implemented by the following execution steps:

1. Given the regional disparity map $\mathfrak{D}^{(R)}$ associated with the left view, compute partition $\tilde{\mathcal{L}_r}$, transferred from partition $\mathcal{L}_l$, using algorithm 6.1.
2. Perform an adaptive erosion of fixed scale $\alpha \in [0; 1[$, on each cell of the transferred partition $\tilde{\mathcal{L}_r}$, so as to produce partition $\tilde{\mathcal{L}_r^\alpha}$. Given that $\mathbf{M}(\mathbf{I}, \alpha)$ is the result of the adaptive erosion of scale α on a binary image $\mathbf{I}$, $\tilde{\mathcal{L}_r^\alpha}$ may be expressed by equation 6.2, for all labels ℓ contained in $\mathcal{L}_l$.
$$\tilde{\mathcal{L}_r^\alpha} = \max_\ell \left\{ \ell \times \mathbf{M}(\bar{T}(\tilde{\mathcal{L}_r} - \ell), \alpha) \right\} \tag{6.2}$$
3. Deduce label image $\mathcal{L}_E^\alpha$ from partition $\tilde{\mathcal{L}_r^\alpha}$ and the adaptively eroded partition $\mathcal{L}_r^\alpha$ of the right view of the stereo pair, so that regions in the right view which are occluded in the left view have a unique marker.
4. Given the gradient of the right view $\mathbf{S}_r$, update $\mathcal{L}_E^\alpha$, such that $\mathcal{L}_E^\alpha[x, y] = 0$ if (x, y) is not a minimum of $\mathbf{S}_r$.
5. Compute the watershed transformation of $\mathbf{S}_r$ using $\mathcal{L}_E^\alpha$ as the initial image of markers, in order to compute the equivalent partition of $\mathcal{L}_l$ for the right view of the stereo pair, where corresponding regions share the same label.

In the experiments performed on low-baseline stereo images and on the Middlebury 2002 benchmark, we found that the erosion scale of $\alpha = 0.25$ produced very good results, some of which are displayed in figure 6.3. It is advisable not to use excessively strong erosions, otherwise the markers may no longer be sufficiently pertinent to describe the region they are intended to segment. Furthermore, the partition of the right image is expected to have the same or a higher degree of coarseness, compared to the left image partition. Indeed, at a finer degree of segmentation, more cells of the right partition are likely to be unmarked by the transferred partition, although these cells do not necessarily correspond to occluded areas in the left view, which is not desirable.

From sparse disparity measures to co-segmentation

Sparse disparity maps computed using the diffusion method introduced in section 5.2, can also be used to generate co-segmentations of stereo images. As usual, the first step of the process consists in transferring labels from the left to the right view of the stereo pair. Algorithm 6.1 can be used to this end, but it is worth mentioning that the process can be greatly simplified. Indeed, by construction, all our sparse disparity measures satisfy the cross-checking criterion. Therefore, any of our sparse disparity maps, say $\mathfrak{D}^{(S)}$, induces a bijection between the correspondences of the left and right views of the stereo pair, ensuring that no overlap will occur while transferring the labels. Unfortunately, this also means that the occlusion mask cannot be generated from the sparse disparity data.

Note that, when using algorithm 6.1, it will be necessary to alter the transfer domain at line

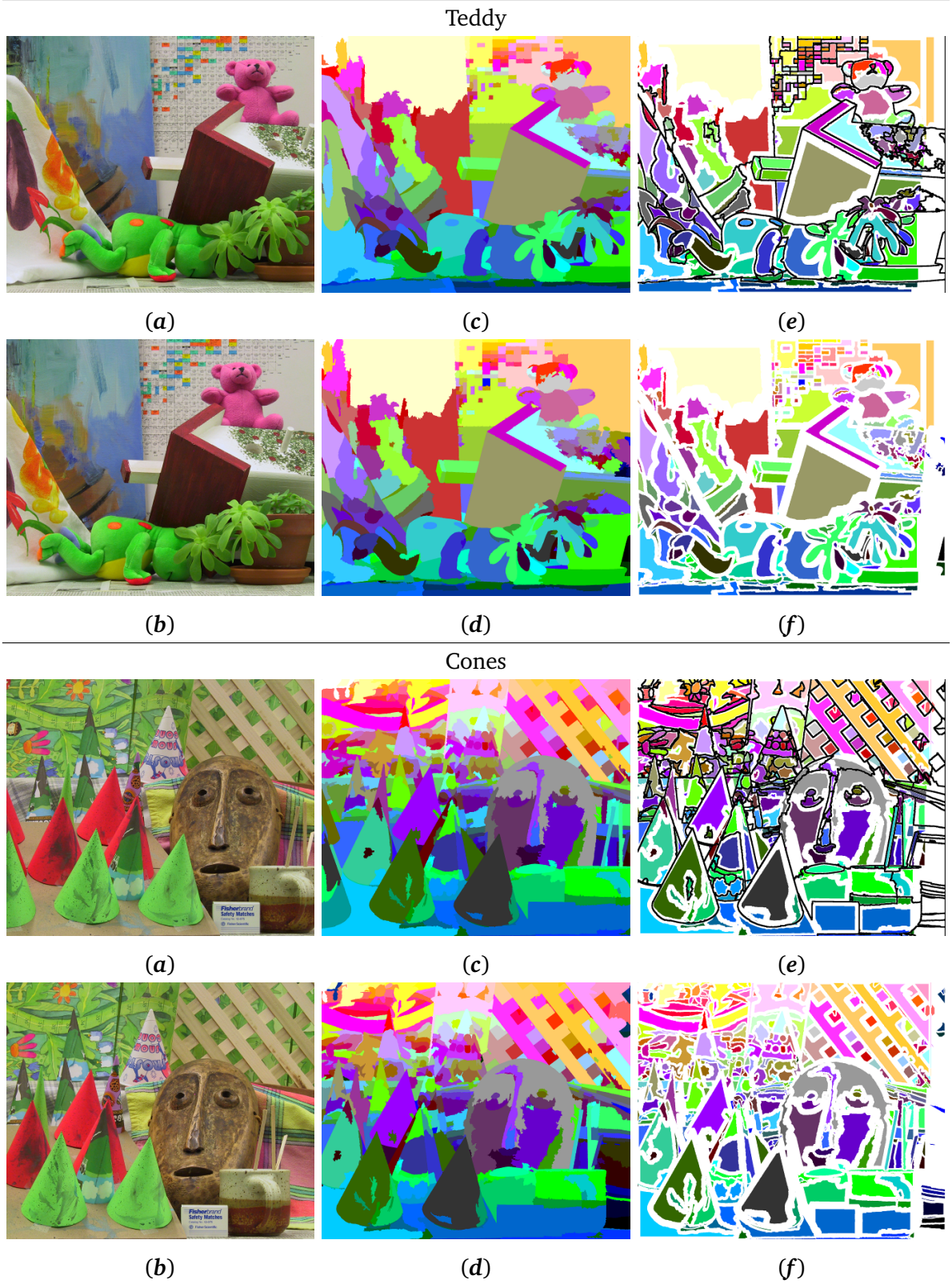


FIGURE 6.3: Generation of image co-segmentations from regional disparities. (a) Left view of the stereo pair. (b) Right view of the stereo pair. (c) The partition of the left view $\mathcal{L}_l$. (d) Co-segmentation of the right view. (e) The partition $\tilde{\mathcal{L}}_r^{\alpha}$ obtained by transferring the labels of $\mathcal{L}_l$ according to regional disparity map $\mathfrak{D}^{(R)}$, and by applying an adaptive erosion to the cells of the transferred partition. The black lines represent the frontiers of the initial right image partition $\mathcal{L}_r$. (f) Regions in $\mathcal{L}_r$ not marked by $\tilde{\mathcal{L}}_r^{\alpha}$ receive new labels.

9, so that (x, y) belongs to the set of points associated with a measure in the provided sparse disparity map, $\mathfrak{D} = \mathfrak{D}^{(S)}$. The rest of the procedure strictly follows the instruction set given for the co-segmentation controlled by regional disparities. Finally, since the sparse measures have a high accuracy, the strength of the adaptive erosion may be reduced: we chose $\alpha = 0.05$ in place of 0.25. Figure 6.4 shows some of the co-segmentations obtained on the Middlebury 2014 dataset. In this experiment, the over-segmentation that was used to generate the sparse disparity maps of the left view, has been re-employed to produce the co-segmentation of the right view.

Known issues and possible ameliorations While the co-segmentations still look consistent, there are some issues related either to the usage of sparse measures or to the complexity of the scene. The first thing to remark is that the co-segmentation will be highly dependent on the efficiency of the pruning of bad measures, as discussed in section 5.2.2. Indeed, one error suffices to transfer a pixel with its label to the wrong catchment basin of the second image gradient. Although this error may concern just one pixel, the effect on the final segmentation is far more significant. This problem may be observed in the `Adirondack` example, where the slat of an armchair has merged with some parts of the background. It is not an issue affecting regional disparities, because of their strong and inherent regional consistency. The second thing to remark is about the complexity of the segmented scenes, for which the ordering constraint does not apply, as is the case for the background visible between the mug and its handle in `Adirondack`. We see that this internal region is shifted correctly to the right image of the stereo pair, therefore we know the part of the background in the right image which corresponds to the area enclosed by the contours of the mug. But this region does not seem meaningful at all with respect to the right image of the stereo pair. As part of a future work, it would be useful to improve two aspects of these co-segmentations:

1. In the generated co-segmentation, suppress segmentation borders along which the gradient of the disparity function never exceeds a specific intensity. Note that this would necessitate filling the holes of the disparity map: a strategy to fulfil this goal will be discussed in the succeeding chapter.
2. Take advantage of the regional merging induced by this border deletion to relabel the regions of the left view, so that both left and right segmentations are synchronised. As a result, a region such as that surrounded by the mug would inherit the label of the background object to which it belongs.

6.2 Applications

Given the perfect co-segmentations of two stereo images, we can prevent matches across the stereo pair of points belonging to different regions. For example, when using the diffusion algorithm

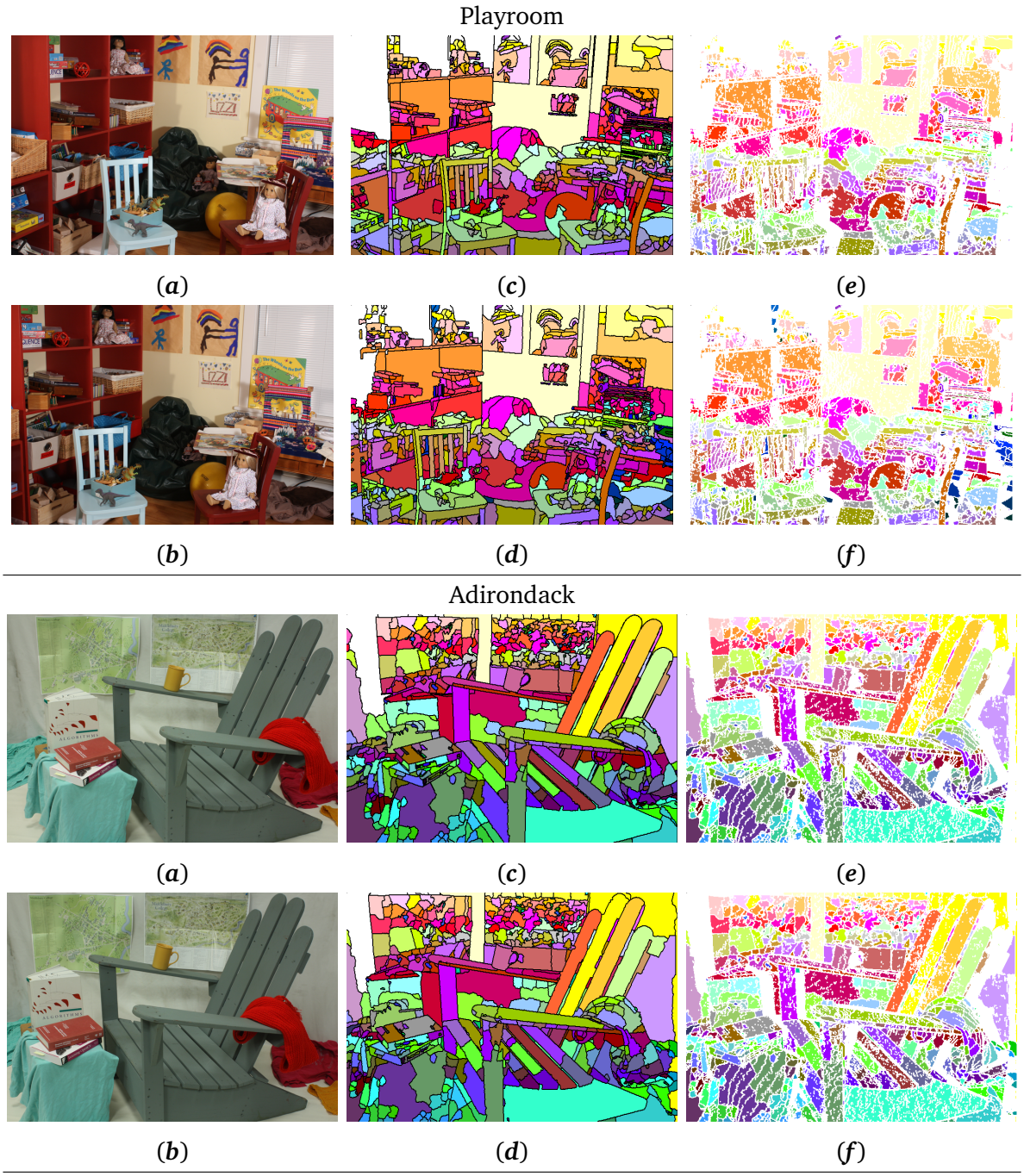


FIGURE 6.4: Generation of image co-segmentations from sparse disparities. (a) Left view of the stereo pair. (b) Right view of the stereo pair. (c) The partition of the left view $\mathcal{L}_l$. (d) Co-segmentation of the right view. (e) The partition $\tilde{\mathcal{L}}_r^{\alpha}$ obtained by transferring the labels of $\mathcal{L}_l$ according to regional disparity map $\mathfrak{D}^{(S)}$, and by applying an adaptive erosion to the cells of the transferred partition. (f) Regions in $\mathcal{L}_r$ not marked by $\tilde{\mathcal{L}}_r^{\alpha}$ receive new labels.

presented in section 5.2, the disparity space volume $\mathbf{D}$, provided as input of the algorithm, should be updated in order to verify that $\mathbf{D}[x, y, d]$ equals the maximum superimposition cost c_{max} when points (x, y) in the left view $\mathbf{I}_l$ and $(x - d, y)$ in the right view $\mathbf{I}_r$ are allocated different labels in their respective equivalent segmentations. Beyond the simple alteration of the DSV, equivalent partitions have further useful applications in image interpolation and stereo matching, two of which are briefly summarised in this section.

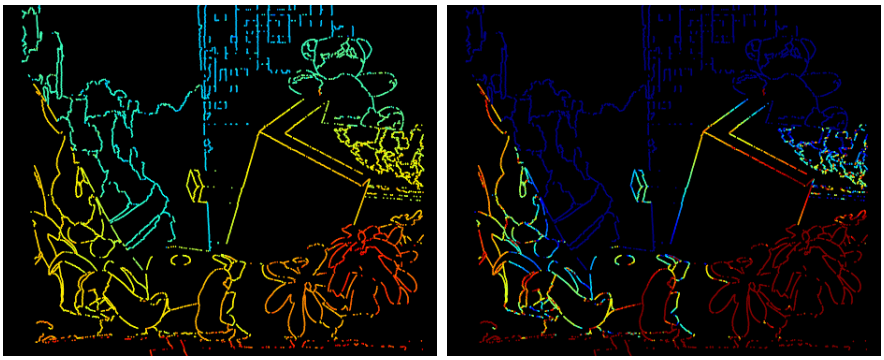
Interpolation of partitions

The interpolation of two equivalent partitions is an extension of the morphological median of two binary images [Beucher, 1998]. Let $\mathcal{L}_0$ and $\mathcal{L}_1$ be two complete and equivalent partitions. We denote by $\mathbf{M}_{0,1}$, the binary image highlighting the points allocated the same label in both partitions, i.e. $\mathbf{M}_{0,1}[x, y] = 1 \Leftrightarrow \mathcal{L}_0[x, y] = \mathcal{L}_1[x, y]$. The median partition of $\mathcal{L}_0$ and $\mathcal{L}_1$ is defined as $\mathcal{L}_{(0+1)/2}$ and is expressed as $\mathcal{L}_{1/2} = \text{W.T.}(\mathbf{S}, \mathcal{L}_0 \times \mathbf{M}_{0,1})$, where W.T. represents the watershed transformation operator, and $\mathbf{S}$ the topographical surface controlling the flooding, such that $\mathbf{S}[x, y] = 0$ for all pixels of the image domain. From this median partition, it is possible to generate new intermediate partitions $\mathcal{L}_{1/4}$ from $\mathcal{L}_0$ and $\mathcal{L}_{1/2}$, and $\mathcal{L}_{3/4}$ from $\mathcal{L}_{1/2}$ and $\mathcal{L}_1$. By iteration, we can ultimately construct a sequence of intermediate partitions between $\mathcal{L}_0$ and $\mathcal{L}_1$ of the form $\mathcal{L}_{1/2^n}, \ldots, \mathcal{L}_{k/2^n}, \mathcal{L}_{(k+1)/2^n}, \ldots, \mathcal{L}_{(2^n-1)/2^n}$, for any $n \in \mathbb{N}$. When each cell is mapped to grey level in function of its label, this sequence of partitions will transform into a sequence of mosaic images estimating the intermediate frames, and thereby the motion which occurred between the two *keyframes* partitioned as $\mathcal{L}_0$ and $\mathcal{L}_1$. This type of mechanism has very attractive features in the context of video compression. In order to be successful, it is necessary that a co-segmentation of the keyframes be used and that an overlap persist with each pair of equivalent cells when $\mathcal{L}_0$ and $\mathcal{L}_1$ are superimposed.

Contour point matching

Co-segmentations may be employed to facilitate the establishment of correspondences between points lying on the contours of the equivalent stereo segmentations. This aspect is interesting when the co-segmentations originate from regional disparity maps, since they do not contain accurate measures. The purpose of contour point matching in this case study therefore is to obtain accurate measures along the contours of the coarse regions.

We assume first, that the images are rectified, so that the corresponding points have the same ordinates in both images, and second, that the ordering constraint applies. For a particular ordinate y, we extract the contour points from both the left and right partitions of the stereo pair, and recover their respective sequences of abscissa: $S_l = \{x_1, \ldots, x_M\}$ and $S_r = \{x'_1, \ldots, x'_N\}$. Both sequences are strictly increasing, i.e. $x_i < x_j$ and $x'_i < x'_j$ for $i < j$. Given the left and right equivalent partitions of the stereo pair, say $\mathcal{L}_l$ and $\mathcal{L}_r$, we define the cost of matching the i-th



(*a*) (*b*)

FIGURE 6.5: Visualisation of the contour disparities obtained for `Teddy` at resolution 1800×1600 pixels using the dynamic time warping procedure. The co-segmentations used to perform this computation are driven by the regional disparity map of the coarse partition, as shown in figure 6.3. The maximum variability with respect to the regional disparity assigned to each point was set to $\tau = 20$ pixels. The jet colormap is used to display disparities in the range of (*a*) $[41; 197]$ pixels and (*b*) $[115; 140]$ pixels.

point in S_l with the j-th point in S_r as $c(x_i, x'_j)$. We impose that $c(x_i, x'_j) = 0$ when the following conditions are satisfied:

1. the labels surrounding the two points on the same scanline are identical, i.e. if $\mathcal{L}_l[x_i - 1, y] = \mathcal{L}_r[x'_j - 1, y]$ and $\mathcal{L}_l[x_i + 1, y] = \mathcal{L}_r[x'_j + 1, y]$.
2. the disparity induced by the matching of the two points, i.e. $x_i - x'_j$ must satisfy $|(x_i - x'_j) - \mathfrak{D}^{(R)}[x_i, y]| \leqslant \tau$, τ being the maximum variability in disparity allowed with respect to the regional disparity allocated to the point in the left view.

Otherwise, we deduce that the two points cannot be matched, in which case $c(x_i, x'_j) = +\infty$.

In order to enforce the ordering constraint, the definitive matches result from the backtracing of a dynamic time warping (cf. section 2.1.4) of the two sequences S_l and S_r. In particular, if $\mathbf{A} : \{0, \ldots, M\} \times \{0, \ldots, N\} \to \mathbb{R}$ represents the accumulator array of the warping costs, the latter is initialised such that $\mathbf{A}[0, 0] = 0$ and $\mathbf{A}[i, j] = +\infty$ if $i > 0$ or $j > 0$. The accumulator array is then recursively updated using the recurrence relationship as follows:

$$\begin{aligned} \mathbf{A}[i, 0] &= \mathbf{A}[i-1, 0] + \xi, \ \forall\, i \geqslant 1 \\ \mathbf{A}[i, j] &= \min \begin{cases} \mathbf{A}[i-1, j-1] + c(x_i, x'_j) \\ \mathbf{A}[i-1, j] + \xi \\ \mathbf{A}[i, j-1] + \xi \end{cases} , \ \forall\, i, j \geqslant 1 \end{aligned}$$

with $\xi > 0$, so that the warping path is encouraged to pass by the pair of points having a null matching cost. Though simple, this procedure results in the contour disparities displayed in figure 6.5 for the `Teddy` example. Current limitations incurred by the chosen matching cost are

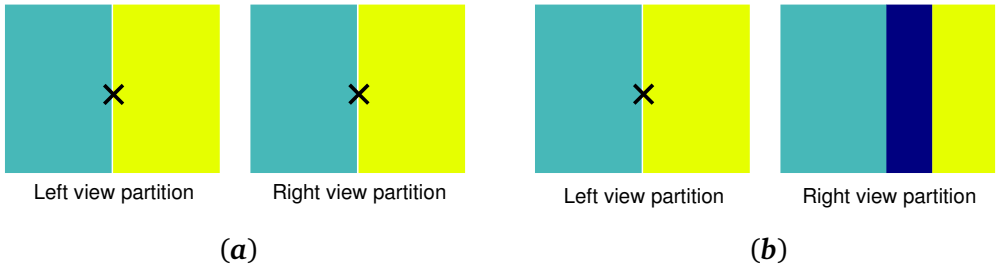


FIGURE 6.6: Comparing the labels surrounding contour points makes sense as long as (a) no region is totally occluded in one view of the stereo pair. (b) Occluded regions change the labelling configuration of the contour points and prevent them from being matched by the proposed algorithm.

the inability to distinguish points lying along horizontal contours, and the absence of a strategy to deal with totally occluded regions, which could change the label coding of some points across the stereo pair, as illustrated in figure 6.6.

Summary

In this chapter, an asymmetric approach to the computation of equivalent stereo partitions has been proposed: given the partition for the left view of the stereo pair, one computes the equivalent partition for the right view. To proceed, it is necessary to separately mark the areas of the right view being occluded in the left view, and to transform the cells of the left partition into markers for the segmentation of the right view. The cells of the left partition are transferred to the right view taking account of disparity information. We investigated the use of regional disparities and sparse disparity maps to serve that purpose and identified the possible pitfalls. What transpired from the experiments was that the regional disparities typically lead to high quality co-segmentations, provided the geometric assumptions, which require satisfaction for the computation of regional disparities, remain satisfied. Additionally, we showed that such co-segmentations provide useful clues to the computation of accurate contour based disparities. This function will be reused in one of our case studies of disparity map estimation. As far as the sparse measurements are concerned, the resulting co-segmentations are globally pertinent, but may comprise some undesirable artefacts. These are mainly due to rare disparity errors being unavoidable in sparse measurements, and to the non-respect of the ordering constraint, for which we suggested a solution.

We are now ready to introduce the final chapter, which concentrates on the most important application of disparity measures: the estimation of the final disparity maps.

Résumé du chapitre 7

Dans ce dernier chapitre, nous abordons les méthodes d'estimation servant à la finalisation des cartes de disparités. Le choix de la méthode dépend de la nature des mesures de disparités considérées.

Par exemple, les mesures de disparité obtenues par diffusions multi-échelles peuvent être considérées comme très fiables. L'estimation de la carte de disparités complète consiste donc à remplir les trous de la carte de disparités sporadique avec des valeurs de disparités plausibles. Dans ce contexte, nous aurons recours à l'interpolation de fonctions basées sur les fonctions distance introduites au chapitre 3. Prenons maintenant les cartes de disparités régionales mesurées à une échelle grossière ainsi qu'à une échelle fine de la segmentation de l'image de référence. Dans ce cas de figure, l'objectif de la méthode d'estimation consiste à corriger les mesures de disparités régionales qui sont erronées vis-à-vis de la partition fine.

Les méthodes d'estimation présentées dans ce chapitre partagent néanmoins des objectifs identiques. La carte de disparités finale doit, au travers de chaque région grossière, évoluer sans discontinuité. Par ailleurs, les mesures fiables doivent être préservées tandis que les mesures erronées doivent être détectées de telle sorte à ne pas perturber le mécanisme d'estimation qui se chargera par la suite de les corriger.

Le chapitre 7 présente les résultats obtenus pour chaque méthode testée. La dernière méthode exposée est appliquée sur l'ensemble de la base de Middlebury 2014, et fait l'objet d'une évaluation plus approfondie dont le lecteur trouvera le détail dans l'annexe.

Chapter 7

Estimation for Depth Map Computation

The estimation phase is an essential aspect of depth map computation. It takes place after the measurement of disparities. In the case of sparse disparities, it comprises filling the holes of the disparity map meaningfully, while in the case of regional disparities, it consists of using available disparity measures to infer more accurate disparity maps.

The chapter is arranged as four independent sections, each presenting an estimation technique with an application relevant to the computation of disparity maps. The first two sections specifically target regional disparity maps: in section 7.1, we will study a linear estimator arising from the field of geostatistics, called "kriging". Used in conjunction with a segmentation of the reference image, a linear variability model and both contour and internal disparity points, kriging can produce very appealing disparity maps. In order to extract internal disparity points reliably, we resort to a regularised version of the regional disparity map calculated at a fine level of segmentation. The regularisation scheme, based on maximum a posteriori estimation is presented in section 7.2. The last two sections concern the filling of sparse disparity maps. We start in section 7.3, with a very simple filling technique based on regional statistics. This will essentially serve to improve the fattening effect removal and therefore, the quality of the multi-scale diffusion. Finally, section 7.4 focuses on the use of distance functions to accomplish the final disparity map interpolation.

7.1 Kriging with linear variograms

Let $\mathbf{S}$ be a topographical surface, such that $\mathbf{S}[x, y]$ denotes the altitude of the relief with respect to the zero-level coordinate (x, y). The problem is the following: given a set of sample data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ for which the altitudes $\mathbf{S}_i = \mathbf{S}[x_i, y_i]$ are known for $i = 1, \ldots, n$, estimate the altitudes of the other points of the relief.

Linear estimation The purpose of a linear estimator is to express the altitudes of the *estimated* topographical surface, as a linear combination of the altitudes provided for the sample points.

Let $\mathbf{S}^\circ$ be the estimated topographical surface, then:

$$\mathbf{S}^\circ[x_0, y_0] = \sum_{i=1}^{n} \lambda_i[x_0, y_0]\mathbf{S}[x_i, y_i]$$

In order to simplify the notation of the above formula, we can write it more compactly as $\mathbf{S}_0^\circ = \sum_{i=1}^{n} \lambda_i \mathbf{S}_i$, so that the weights $\lambda_1, \ldots, \lambda_n$ control only the linear combination allocated to the estimation of (x_0, y_0). In order to perform this estimation using a kriging operator and to understand what it does, it is useful to view $\mathbf{S}$ as the *realisation* of a random process $\mathcal{F}$. The kriging operator seeks a linear estimator $\mathcal{F}^\circ$ in the same way as that above, as $\mathcal{F}_0^\circ = \sum_{i=1}^{n} \lambda_i \mathcal{F}_i$. It is defined such that the following properties hold:

1. The estimator is unbiased. That is: $\mathbb{E}\left[\mathcal{F}_0^\circ - \mathcal{F}_0\right] = 0$
2. The variance of the estimation error, $\mathrm{Var}\left(\sum_{i=1}^{n} \lambda_i \mathcal{F}_i - \mathcal{F}_0\right)$, is minimised.

The problem of kriging therefore comprises finding the weights $\lambda_1, \ldots, \lambda_n$ which satisfy these two properties.

Kriging: a solution Let us provide a solution for the case where, for any point of the relief denoted by index k, $\mathbb{E}[\mathcal{F}_k] = \mathbb{E}[\mathcal{F}_k^\circ] = m$, where m represents a constant real number. This particular case is often referred to as *ordinary kriging*. We wish to minimise the variance of the estimation error, under the non-bias constraint. The constraint states that:

$$\begin{aligned}
\mathbb{E}[\mathcal{F}_0^\circ - \mathcal{F}_0] &= \sum_{i=1}^{n} (\lambda_i \mathbb{E}[\mathcal{F}_i]) - \mathbb{E}[\mathcal{F}_0] \\
&= m\left(\sum_{i=1}^{n} \lambda_i - 1\right) \\
&= 0
\end{aligned}$$

Therefore, we deduce that: $\sum_{i=1}^{n} \lambda_i = 1$. The objective function to minimise is then expressed as:

$$\Phi(\lambda_1, \ldots, \lambda_n, \mu) = \mathrm{Var}(\mathcal{F}_0^\circ - \mathcal{F}_0) + 2\mu\left(\sum_{i=1}^{n} \lambda_i - 1\right)$$

The intuition behind this objective function is that, if the constraint is verified, i.e. if $\sum_{i=1}^{n} \lambda_i - 1 = 0$, then it is indeed the variance of the estimation error which is minimised. In calculus, μ corresponds to a Lagrange multiplier. We now need to find the parameters minimising Φ, which can be achieved by solving the following system of equations:

$$\begin{aligned}
\frac{\partial}{\partial \lambda_i}\Phi &= 0 \text{ for all } i \in 1, \ldots, n \\
\frac{\partial}{\partial \mu}\Phi &= 0
\end{aligned}$$

To proceed, it is necessary to recall some relations of variance algebra:

$$\mathrm{Var}\left(\sum_{i=1}^{n}\lambda_i\mathcal{F}_i\right) = \mathbb{E}\left[\left(\sum_{i=1}^{n}\lambda_i\mathcal{F}_i\right)^2\right] - \left(\mathbb{E}\left[\sum_{i=1}^{n}\lambda_i\mathcal{F}_i\right]\right)^2$$

The square of a sum may be simplified as follows:

$$\begin{aligned}\left(\sum_{i=1}^{n}a_i\right)^2 &= a_1^2 + 2a_1\sum_{i=2}^{n}a_i + \left(\sum_{i=2}^{n}a_i\right)^2\\ &= a_1^2 + 2a_1\sum_{i=2}^{n}a_i + \ldots + a_{n-1}^2 + 2a_{n-1}a_n + a_n^2\\ &= \sum_{i=1}^{n}a_i^2 + 2\sum_{j=1}^{n-1}\sum_{i=j+1}^{n}a_ia_j\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}a_ia_j\end{aligned}$$

Therefore,

$$\begin{aligned}\mathrm{Var}\left(\sum_{i=1}^{n}\lambda_i\mathcal{F}_i\right) &= \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j\mathcal{F}_i\mathcal{F}_j\right] - \left(\sum_{i=1}^{n}\lambda_i\mathbb{E}\left[\mathcal{F}_i\right]\right)^2\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j\mathbb{E}\left[\mathcal{F}_i\mathcal{F}_j\right] - \sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j\mathbb{E}\left[\mathcal{F}_i\right]\mathbb{E}\left[\mathcal{F}_j\right]\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j\mathrm{Cov}\left(\mathcal{F}_i,\mathcal{F}_j\right)\end{aligned}$$

Given that the variance of a linear combination of two random variables X and Y may be derived from the above equation as $\mathrm{Var}(aX+bY) = a^2\mathrm{Var}(X) + b^2\mathrm{Var}(Y) + 2ab\mathrm{Cov}(X,Y)$, we deduce that:

$$\begin{aligned}\mathrm{Var}\left(\sum_{i=1}^{n}\lambda_i\mathcal{F}_i - \mathcal{F}_0\right) &= \mathrm{Var}\left(\mathcal{F}_0\right) + \mathrm{Var}\left(\sum_{i=1}^{n}\lambda_i\mathcal{F}_i\right) - 2\mathrm{Cov}\left(\sum_{i=1}^{n}\lambda_i\mathcal{F}_i,\mathcal{F}_0\right)\\ &= \mathrm{Var}\left(\mathcal{F}_0\right) + \sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j\mathrm{Cov}\left(\mathcal{F}_i,\mathcal{F}_j\right) - 2\sum_{i=1}^{n}\lambda_i\mathrm{Cov}\left(\mathcal{F}_i,\mathcal{F}_0\right)\end{aligned}$$

We can now find the parameters minimising the objective function Φ, by solving the following system of linear equations:

$$\begin{aligned}\frac{\partial}{\partial\lambda_i}\Phi = 0 &= 2\sum_{j=1}^{n}\lambda_j\mathrm{Cov}\left(\mathcal{F}_i,\mathcal{F}_j\right) - 2\mathrm{Cov}\left(\mathcal{F}_i,\mathcal{F}_0\right) + 2\mu\\ \frac{\partial}{\partial\mu}\Phi = 0 &= \sum_{i=1}^{n}\lambda_i - 1\end{aligned}$$

In matrix form, this system of equation may be expressed as equation 7.1:

$$\left[\begin{array}{ccc|c} \mathrm{Cov}(\mathcal{F}_1,\mathcal{F}_1) & \cdots & \mathrm{Cov}(\mathcal{F}_1,\mathcal{F}_n) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathrm{Cov}(\mathcal{F}_n,\mathcal{F}_1) & \cdots & \mathrm{Cov}(\mathcal{F}_n,\mathcal{F}_n) & 1 \\ \hline 1 & \cdots & 1 & 0 \end{array}\right] \left[\begin{array}{c} \lambda_1 \\ \vdots \\ \lambda_n \\ \hline \mu \end{array}\right] = \left[\begin{array}{c} \mathrm{Cov}(\mathcal{F}_0,\mathcal{F}_1) \\ \vdots \\ \mathrm{Cov}(\mathcal{F}_0,\mathcal{F}_n) \\ \hline 1 \end{array}\right] \tag{7.1}$$

Or more compactly, as

$$\left[\begin{array}{c|c} \mathbf{C} & \mathbf{1} \\ \hline \mathbf{1}^\intercal & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\lambda} \\ \hline \mu \end{array}\right] = \left[\begin{array}{c} \mathbf{c}_0 \\ \hline 1 \end{array}\right]$$

The solution of this system can be computed simultaneously for different points for which we seek an estimate: it suffices to replace the equation of the form $\mathbf{A}\boldsymbol{x} = \mathbf{b}$ by $\mathbf{AX} = \mathbf{B}$, such that each column of $\mathbf{A}$ and $\mathbf{B}$ reflects, for a particular target, the sought coefficients and the covariances between data and target respectively. $\mathbf{X}$ can then be found by Gaussian elimination. A more efficient method of computing the altitudes of $\mathbf{S}^\circ$ directly is provided by the dual kriging.

Dual kriging The estimation of the altitude taken at point (x_0, y_0) may be expressed as an inner product of the form

$$S_0^\circ = \left[\begin{array}{ccc|c} S_1 & \cdots & S_n & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\lambda} \\ \hline \mu \end{array}\right]$$

which can be developed via equation 7.1 as:

$$S_0^\circ = \left[\begin{array}{ccc|c} S_1 & \cdots & S_n & 0 \end{array}\right] \left[\begin{array}{c|c} \mathbf{C} & \mathbf{1} \\ \hline \mathbf{1}^\intercal & 0 \end{array}\right]^{-1} \left[\begin{array}{c} \mathbf{c}_0 \\ \hline 1 \end{array}\right] = \mathbf{k}^\intercal \left[\begin{array}{c} \mathbf{c}_0 \\ \hline 1 \end{array}\right]$$

The vector $\mathbf{k}$ depends solely on the kriging configuration; that is the sample points and the covariance between these sample points. This means it can be re-utilised to perform the estimation at any other point of the relief. Now, in order to use the kriging operator properly, it is necessary to determine the model of the random process $\mathcal{F}$.

Intrinsic random processes A random process $\mathcal{F}$ is said to be *intrinsic*, when its *variations* are stationary of order 2, and of null expectation. Mathematically, an intrinsic random process is characterised by the following properties:

$$\mathbb{E}\left[\mathcal{F}[\mathbf{p}_i + \mathbf{h}] - \mathcal{F}[\mathbf{p}_i]\right] = 0 \tag{7.2}$$

$$\begin{aligned} \mathrm{Var}\left(\mathcal{F}[\mathbf{p}_i + \mathbf{h}] - \mathcal{F}[\mathbf{p}_i]\right) &= \mathbb{E}\left[\left(\mathcal{F}[\mathbf{p}_i + \mathbf{h}] - \mathcal{F}[\mathbf{p}_i]\right)^2\right] \\ &= 2 \cdot \Upsilon(\|\mathbf{h}\|) \end{aligned} \tag{7.3}$$

Equation 7.2 indicates that we should expect no particular tendency of variation in altitude when shifting the points of the relief in a particular direction $\mathbf{h}$, while equation 7.3 imposes the condition that the variance of the differences in altitude at any two points be stationary, i.e. invariant to translation. Υ denotes the *variogram*, which models the variability of altitudes, taken at two points, in function of their distance $\|\mathbf{h}\|$. When using intrinsic processes, we can replace the covariance terms in equation 7.1 with the additive inverse of the values taken by the variogram Υ in function of the spatial distance between the points we are considering. We refer the reader to [Linchtenstern, 2013] for a proof of this assertion. When the covariance of $\mathcal{F}$ is also stationary, the relationship between the variogram and the covariance is given by the following relation:

$$\begin{aligned}
\Upsilon(\|\mathbf{h}\|) &= \frac{1}{2} \cdot \mathrm{Var}(\mathcal{F}_i - \mathcal{F}_j) \\
&= \frac{1}{2} \left(\mathrm{Var}(\mathcal{F}_i) + \mathrm{Var}(\mathcal{F}_j) - 2\mathrm{Cov}(\mathcal{F}_i, \mathcal{F}_j)\right) \\
&= \frac{1}{2} \left(\mathrm{Cov}(\|\mathbf{p}_i - \mathbf{p}_i\|) + \mathrm{Cov}(\|\mathbf{p}_j - \mathbf{p}_j\|) - 2\mathrm{Cov}(\|\mathbf{p}_i - \mathbf{p}_j\|)\right) \\
&= \mathrm{Cov}(0) - \mathrm{Cov}(\|\mathbf{h}\|)
\end{aligned}$$

$\mathbf{h}$ denoting the distance between points $\mathbf{p}_i$ and $\mathbf{p}_j$. A comprehensive survey of variogram models is provided in [Delhomme, 1976]. In geostatistics, the parameters of the chosen model are determined experimentally, in order to be in accordance with the type of the topographical surface or phenomenon being studied. When processing images of unknown nature, it is usually more difficult to characterise precisely the process that should result from the interpolator, and thus the variability model needs to be sufficiently general for the kriging to perform well in most scenarios. It is then often desirable to leave the variogram unbounded so that a higher distance between two coordinates of a relief results in a lower correlation of the values (altitude, grey level, disparity, etc.) to which they are attributed. Kriging has already been used in the context of image and video sequence coding [Decenciere et al., 1998], where the variogram followed the thin-plate elastic model. In this work, we shall use the linear variogram for our interpolation requirements.

7.1.1 Application: disparity maps from feature points

A linear variogram is expressed according to the distance between two coordinates of a random function as:

$$\Upsilon(\mathbf{p}_i, \mathbf{p}_j) \propto \|\mathbf{p}_i - \mathbf{p}_j\| \tag{7.4}$$

In this application, we propose to utilise the ordinary kriging operator with a linear variogram so as to interpolate disparity maps from sample points, given an associated regional disparity map.

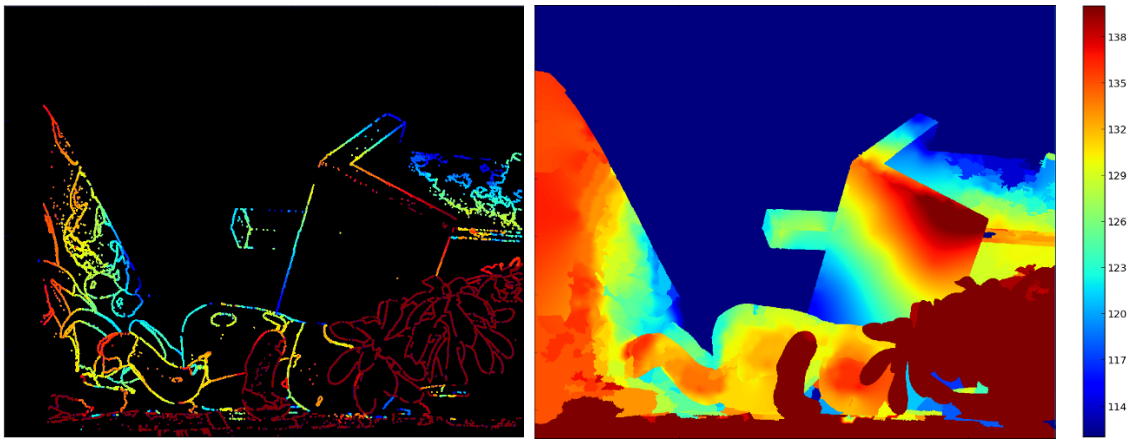
Sample points originate both from the disparities available along the contours of the reference image partition, obtained using the method presented in section 6.2, and from feature points lying in the interior of the partition cells.

Characterisation of internal feature points Candidate internal points are extracted from both images of the stereo pair, using the white and the black top-hat transformations (cf. equation 3.7), which respectively recover the peaks and the cavities of the topographical surface representing the grey-level function attributed to each of the stereo images. These points are matched across the stereo pair using a patch-based correlation and only if cross-checking is satisfied. Only the candidate points being matched with a disparity equal to that of the regional disparity may be part of the kriging sample points. We can regard each of these samples as an anchor of the regional disparity allotted to the related region.

Sample point selection Let $\mathcal{L}_l$ be the partition of the reference image. S denotes the set of sample points $\{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$, such that each pixel $\mathbf{p}_i \in S$ is associated with a disparity measure $\mathfrak{D}[\mathbf{p}_i]$. We now seek the estimated disparity $\mathfrak{D}^\circ[\mathbf{p}_0]$ of point $\mathbf{p}_0$ using the kriging operator controlled by linear variogram. Employing a linear variogram and all the sample points in S means that there would always be a dependency between $\mathfrak{D}^\circ[\mathbf{p}_0]$ and $\mathfrak{D}[\mathbf{p}_i]$ for any $i \in \{1, \ldots, n\}$. This is of course not realistic in the context of disparity map estimation. Therefore, the set of points we need to consider in order to compute $\mathfrak{D}^\circ[\mathbf{p}_0]$ must be a subset of S, say $S_0 \subseteq S$. Furthermore, the neighbourhood of $\mathbf{p}_0$ inside which we expect some correlation between the disparity values taken in $\mathfrak{D}$, extends to the frontiers of the region to which $\mathbf{p}_0$ belongs. Hence, $S_0 \subseteq \{\mathbf{p}_i \in S \mid \mathcal{L}_l[\mathbf{p}_0] = \mathcal{L}_l[\mathbf{p}_i]\}$. This is equivalent to stating that the regional boundaries represent geological faults in the topographical surface associated with the disparity function $\mathfrak{D}$. Finally, it should be noted that the contour points considerably outnumber the internal points and that choosing two sample points in close proximity, such as those available along the contours, could lead to numerical instabilities when solving the kriging system of equation 7.1. It is therefore important to ensure that the sample points included in S_0 are spatially well distributed. If all these constraints are taken into consideration, the kriging can produce piece-wise smooth disparity maps such as that displayed in figure 7.1.

7.2 Maximum a posteriori estimation for Markov Random Fields

If we wish to extract more internal points to feed the kriging system described in the previous section, it may be helpful to substitute the coarse regional disparity map for one obtained at a finer degree of segmentation. In section 5.1, we saw that the regional disparities obtained at a fine scale of segmentation were globally accurate, but that some regions were allocated wholly invalid measures. The purpose of this section is to introduce an estimation mechanism which will view the regional disparities obtained both at the coarse and at the fine level of a segmentation,



(*a*) (*b*)

FIGURE 7.1: Kriging with linear variogram on the `Teddy` test case. (*a*) Representation of the sample points providing the disparity data. (*b*) Output of the kriging applied separately to each cell of the coarse partition accompanying `Teddy`.

as *observations*. These observations will in turn be analysed to deduce a regional disparity map, where each set of fine regions contained in the same coarse region, constitutes a Markov field.

Markov random fields A Markov Random Field, or MRF, is a particular kind of undirected graph $G = (V, E)$, composed of a set of nodes V and edges E, for which there exists a transformation $\mathcal{F}$ mapping each node $v \in V$ to a random variable $\mathcal{F}(v)$. The edges of G model the dependencies between these random variables as follows:

- $\mathcal{F}(v_i)$ is independent of $\mathcal{F}(v_j)$ if there is no path in G from v_i to v_j
- Given the set of random variables associated with the points lying in the direct neighbourhood of $v_i \in V$, i.e. $\{\mathcal{F}(v_k) \mid (v_k, v_j) \in E\}$, $\mathcal{F}(v_i)$ is independent of $\mathcal{F}(v_j)$ if $(v_i, v_j) \notin E$.

Assuming that G is fully connected, the MRF is characterised as *pairwise* when the joint distribution of the random variables associated with all the nodes $v_1, \ldots, v_n \in V$, factorise as:

$$\Pr\{\mathcal{F}(v_1), \ldots, \mathcal{F}(v_n)\} = \frac{1}{Z} \prod_{(i,j) \mid (v_i, v_j) \in E} \phi_{i,j}(\mathcal{F}(v_i), \mathcal{F}(v_j)) \tag{7.5}$$

which is also known as a Gibbs distribution with pairwise potential terms. Z is the constant so that $\Pr\{\mathcal{F}(v_1), \ldots, \mathcal{F}(v_n)\}$ denotes a valid probability distribution, and $\phi_{i,j}$ represents the pairwise potential related to the joint probability of random variables $\mathcal{F}(v_i)$ and $\mathcal{F}(v_j)$.

Pairwise MRFs in MAP inference Let $S_1, \ldots, S_n$ be the realisation of the random variables $\mathcal{F}(v_1), \ldots, \mathcal{F}(v_n)$ characterising the MRF described by graph G. At each node $v_i \in V$, we are provided with an observation $\mathcal{O}(v_i)$ of the state S_i taken by $\mathcal{F}(v_i)$. We define the likelihood that we observe event $\mathcal{O}(v_i)$ when $\mathcal{F}(v_i) = S_i$ as: $\Pr\{\mathcal{O}(v_i) \mid \mathcal{F}(v_i) = S_i\}$. The maximum a posteriori inference comprises finding the realisation $S_1, \ldots, S_n$ of $\mathcal{F}$ which maximises the

posterior probability function $\Pr\{\mathcal{F}(v_1),\ldots,\mathcal{F}(v_n) \mid \mathcal{O}(v_1),\ldots,\mathcal{O}(v_n)\}$. Using Bayes' rule and taking the negative logarithm of this objective function, [Prince, 2012] shows that this amounts to solving equation 7.6:

$$S_1,\ldots,S_n = \arg\min_{S_1,\ldots,S_n}\left\{\sum_{i \mid v_i \in V} U_i(S_i) + \sum_{(i,j) \mid (v_i,v_j)\in E} P_{i,j}(S_i,S_j)\right\} \tag{7.6}$$

where the unary and pairwise terms, being identical to those in equation 2.8, are defined as:

$$\begin{aligned} U_i(S_i) &= -\log\left(\Pr\{\mathcal{O}(v_i) \mid \mathcal{F}(v_i) = S_i\}\right) \\ P_{i,j}(S_i,S_j) &= -\log\left(\phi_{i,j}(\mathcal{F}(v_i) = S_i, \mathcal{F}(v_j) = S_j)\right) \end{aligned}$$

In order to solve equation 7.6, we have implemented and employed the method proposed in [Prince, 2012] consisting of labelling the nodes of graph G by means of a max-flow/min-cut algorithm [Kleinberg and Tardos, 2006]. The method is valid for pairwise MRFs containing *discrete* random variables with a finite number of states, and pairwise terms satisfying the *submodularity condition*. We refer the reader to [Prince, 2012] pp. 293–296 for additional details.

7.2.1 Application: regional disparity maps refinement

In section 5.1, we presented a method of computing regional disparities based on the brightness of the stereo images. Figure 5.4(***b***) revealed that the regional disparities obtained at a fine scale of segmentation are generally fairly accurate, but can be subject to serious errors, especially when the regions concerned are occluded. In order to determine whether or not a fine region is occluded, we proposed to employ algorithm 6.1 to compute an occlusion mask from the more approximate regional disparities obtained at the coarse level of segmentation. It is now time to investigate how this information may be exploited to refine the regional disparity maps obtained at the fine level of segmentation. We are provided with the following data:

- $\mathcal{L}_l^{(C)}$ and $\mathcal{L}_l^{(F)}$, denote the coarse and fine partitions of the reference image respectively.
- $\mathbf{R}_1,\ldots,\mathbf{R}_n$, represent the n regions of partition $\mathcal{L}_l^{(F)}$.
- $\mathbf{R}_1^C,\ldots,\mathbf{R}_m^C$, represent the $m \leqslant n$ regions of partition $\mathcal{L}_l^{(C)}$.
- A transformation $\mathbf{H}$ mapping each region $\mathbf{R}_i$ of the fine partition, to the region $\mathbf{H}_\uparrow(\mathbf{R}_i)$ of the coarse partition inside which $\mathbf{R}_i$ is enclosed.
- The *measured* regional disparity for any region $\mathbf{R}_i$, denoted as $d(\mathbf{R}_i)$.
- The *measured* regional disparity for any region $\mathbf{R}_i^C$, denoted as $d(\mathbf{R}_i^C)$.

For each region $\mathbf{R}_i$, we seek the *refined* regional disparity, expressed as $d^\circ(\mathbf{R}_i)$.

MRF characterisation For each region $\mathbf{R}_k^C$ of the coarse partition $\mathcal{L}_l^{(C)}$, there is a set $\mathcal{S}_k$ of regions belonging to the fine partition, such that $\mathcal{S}_k = \{\mathbf{R}_i \mid \forall\, i \in \{1,\dots,n\},\ \mathbf{H}_\uparrow(\mathbf{R}_i) = \mathbf{R}_k^C\}$. We assume that, across each coarse region, the *refined* regional disparity function is the realisation of a Markov Random Field. For a given region $\mathbf{R}_k^C$, the MRF is defined such that each node of the graph corresponds to a region belonging to $\mathcal{S}_k$ and such that its edges model the adjacency relations between the regions of $\mathcal{S}_k$. The definition of the pairwise potentials is what controls the random field. We expect the refined disparities taken by two adjacent nodes of the MRF to be rather close. For that reason, we express the pairwise term of the MRF as:

$$P_{i,j}(d^\circ(\mathbf{R}_i), d^\circ(\mathbf{R}_j)) \propto \left(d^\circ(\mathbf{R}_i) - d^\circ(\mathbf{R}_j)\right)^2 \tag{7.7}$$

This pairwise term therefore models the smoothness of the refined disparity function.

Likelihood characterisation Now the regional disparities *measured* for $\mathbf{R}_k^C$ and for all the regions of the fine partition included in $\mathcal{S}_k$ should be interpreted as observations of the realisation of the MRF described above. Therefore, for a given node of the MRF, the observation consists of the regional disparity allotted to the fine region to which the node corresponds, say $\mathbf{R}_i$, as well as the regional disparity of region $\mathbf{R}_k^C$. Furthermore, we accompany this observation with a binary indicator variable α_i, such that $\alpha_i = 1$ if and only if $\mathbf{R}_i$ is not occluded according to the occlusion mask computed from the coarse regional disparity map. A disagreement between the measured disparity $d(\mathbf{R}_i)$ and the actual disparity $d^\circ(\mathbf{R}_i)$ is possible if $\alpha_i = 0$. This will need to be reflected in the definition of the likelihood. Nonetheless, it is expected that $d^\circ(\mathbf{R}_i)$ remains at a reasonable distance from $d(\mathbf{R}_k^C)$. Therefore, we determine the unary term of equation 7.6 as:

$$U_i(d^\circ(\mathbf{R}_i)) = \alpha_i \left|d^\circ(\mathbf{R}_i) - d(\mathbf{R}_i)\right| + (1-\alpha_i)\left|d^\circ(\mathbf{R}_i) - d(\mathbf{H}_\uparrow(\mathbf{R}_i))\right| \tag{7.8}$$

In other words, when region $\mathbf{R}_i$ is occluded, $d(\mathbf{R}_k^C)$ is used in place of $d(\mathbf{R}_i)$ as a pertinent observation of the realisation $d^\circ(\mathbf{R}_i)$.

Concluding remarks Using the unary and pairwise terms described in this subsection, the solution of equation 7.6 yields the refined regional disparity maps displayed in figure 7.2. Although the choice of the energy terms produces good results, a more rigorous utilisation of the MAP inference for pairwise MRFs should ensure that these energy terms truly represent valid probability functions.

7.3 Hole filling strategy based on regional statistics

The final part of this chapter concerns a strategy for filling the holes which appear in the sparse disparity maps, computed using the cost diffusion algorithm. In this section, we propose a simple

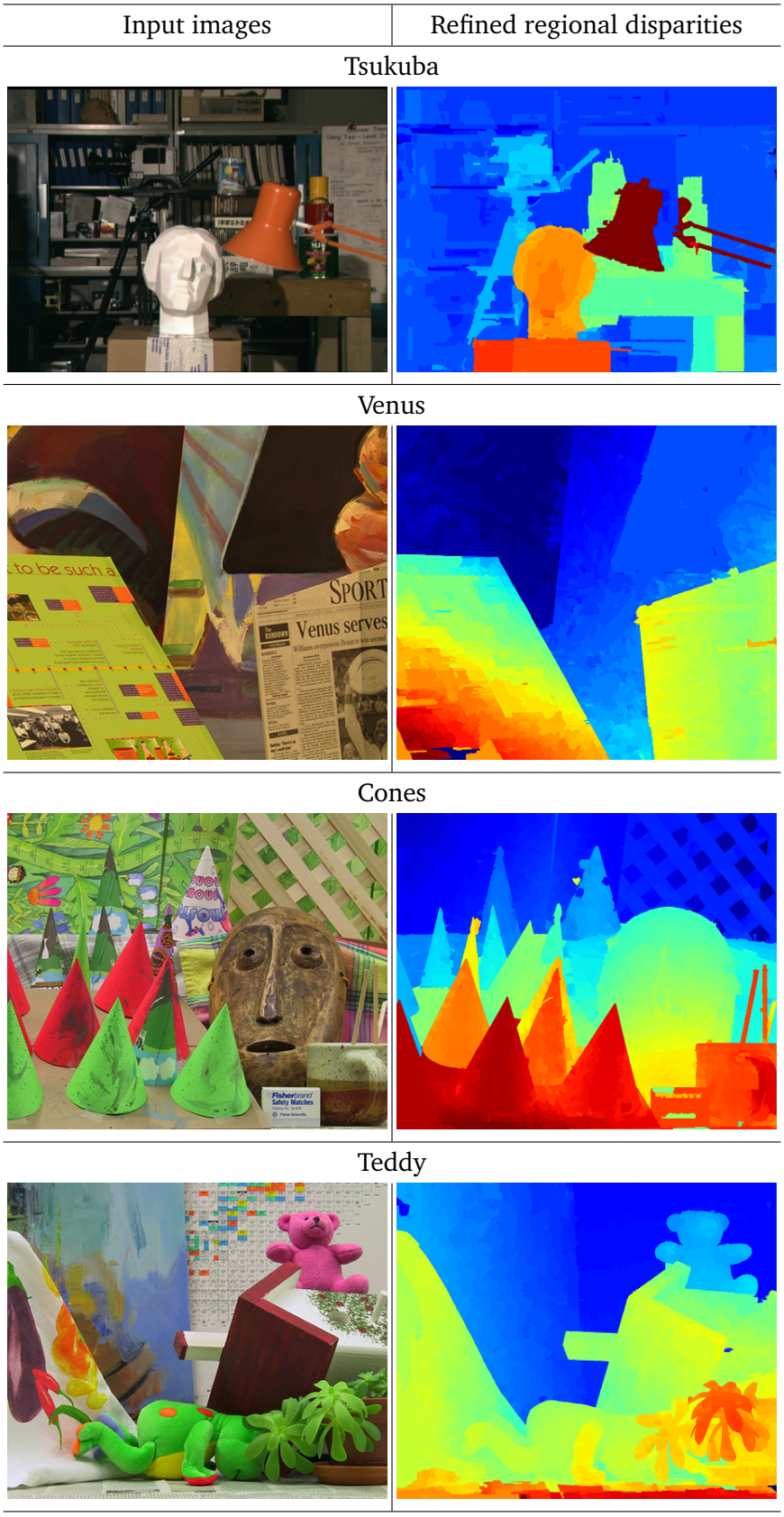


FIGURE 7.2: Regional disparity maps obtained at a fine level of segmentation, using MAP inference across pairwise MRFs. The images showcased in this example originate from Middlebury 2002 dataset.

hole filling strategy, which resorts to look-up-tables generated for each cell of the reference image partition. First, let:

- $\mathfrak{D}^{(S)}$ be a sparse disparity map provided as input of the algorithm. We assume that this sparse disparity map has undergone the cluster-based filtering presented in section 5.2.2.
- X be the set of points having a disparity measure in $\mathfrak{D}^{(S)}$, i.e. $\mathfrak{D}^{(S)}[x, y]$ is defined if and only if $(x, y) \in X$. $\bar{X}$ represents the set of points which do not have a measure in $\mathfrak{D}^{(S)}$.
- $\mathcal{L}$ be the partition of the image for which $\mathfrak{D}^{(S)}$ has been computed. We call $\mathbf{R}_i$ the set of points satisfying the relation $\mathbf{R}_i = \{(x, y) \mid \mathcal{L}[x, y] = i\}$, and thus corresponding to the i-th region of partition $\mathcal{L}$.

For any i-th cell of the partition, we can compute a normalised histogram function h_i mapping a disparity d to its occupancy ratio within $\mathbf{R}_i$. More precisely, if $\bar{T}$ denotes the binary indicator function such that $\bar{T}(x) = 1 \Leftrightarrow x = 0$, then:

$$h_i[d] = \begin{cases} 0 & \text{if } |X \cap \mathbf{R}_i| = 0 \\ \left(\sum_{(x,y) \in X \cap \mathbf{R}_i} \bar{T}\left(\mathfrak{D}^{(S)}[x, y] - d\right) \right) \div |X \cap \mathbf{R}_i| & \text{if } |X \cap \mathbf{R}_i| > 0 \end{cases} \tag{7.9}$$

The cumulative histogram of disparities H_i associated with region $\mathbf{R}_i$ is simply expressed as:

$$H_i[d] = \sum_{k=0}^{d} h_i[k] \tag{7.10}$$

from which we define the q-th upper percentile of h_i as:

$$\mathcal{P}(q, i) = \inf \left\{ d \in \mathbb{N} \mid H_i[d] > \frac{q}{100} \right\} \tag{7.11}$$

The purpose of the hole filling method is to allocate a disparity to every point (x, y) left without a disparity measure in $\mathfrak{D}^{(S)}$, i.e. to any $(x, y) \in \bar{X}$, and under the condition that the region to which the point belongs, contains some disparity measure. In other words, a new disparity is allocated to (x, y) if and only if: $(x, y) \in \bar{X}$ and $(x, y) \in \mathbf{R}_i$, such that $\mathbf{R}_i \cap X \neq \emptyset$. The disparity value allocated to point (x, y) is given by a percentile of h_i, with $q < 100$, so that $\mathcal{P}(q, i)$ corresponds to a disparity measure, within the same region, already belonging to $\mathfrak{D}^{(S)}$. Therefore, suppose that $\mathfrak{D}_q^{(SF)}$ represents the filled in disparity map, then:

$$\mathfrak{D}_q^{(SF)}[x, y] = \begin{cases} \mathfrak{D}^{(S)}[x, y] & \text{if } (x, y) \in X \\ \mathcal{P}(q, i) & \text{if } (x, y) \in \left(\bar{X} \cap \mathbf{R}_i\right) \text{ and } \mathbf{R}_i \cap X \neq \emptyset \end{cases} \tag{7.12}$$

We can also project the percentiles onto each cell of the partition, by computing a disparity map $\mathfrak{D}_q^{(\mathcal{P})}$ using equation 7.13. This regional disparity map will prove useful in the application

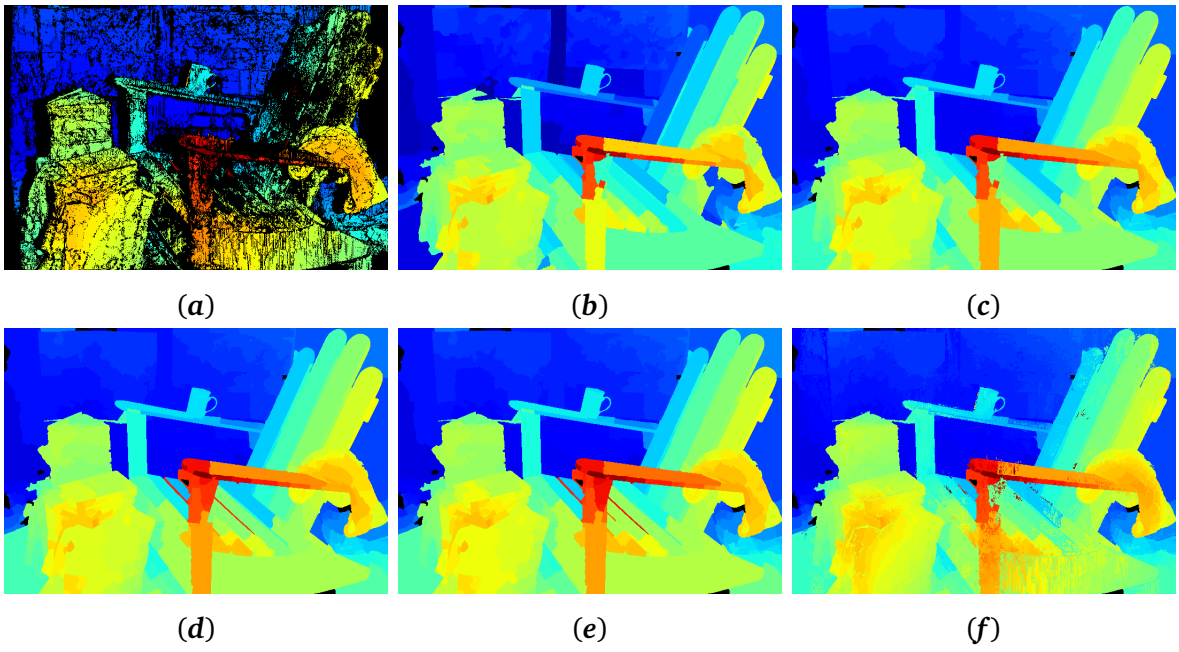


FIGURE 7.3: Estimated disparity maps using hole filling for the `Adirondack` test case. (***a***) Sparse disparity map $\mathfrak{D}^{(S)}$. Regional disparity maps $\mathfrak{D}_q^{(\mathcal{P})}$ obtained for the upper percentiles (***b***) $q = 0$, (***c***) $q = 10$, (***d***) $q = 20$ and (***e***) $q = 30$. Filled-in disparity map $\mathfrak{D}_q^{(SF)}$ for $q = 10$.

presented at the end of this section.

$$\mathfrak{D}_q^{(\mathcal{P})}[x, y] = \mathcal{P}(q, i) \text{ if } (x, y) \in \mathbf{R}_i \text{ and } \mathbf{R}_i \cap X \neq \emptyset \tag{7.13}$$

If we set $q = 0$, then $\mathcal{P}(0, i)$ returns the smallest disparity measure found inside the i-th cell of partition $\mathcal{L}$. We show the effect of this setting in figure 7.3. The majority of the areas which were devoid of disparity measures, are filled in with perceptually satisfying disparity values. This is not surprising, since we know that the missing disparity measures are essentially caused by homogeneous regions or occlusions. In the case of the latter, it is safer to reuse the smallest disparity measure observed across the region than to use the highest, as this high disparity could be due to a fattening artefact. It is, however, good practice to consider the 10-th, the 20-th or even the 30-th percentiles rather than the minimum disparity value found within the region. This will avoid erroneous measures with the smallest disparity values; an occurrence which would deteriorate the quality of the hole filling.

These disparity maps alone should not constitute the final output of a depth estimation algorithm, because they are, as are the regional disparities, inaccurate. But, when comparing image (***a***) to images (***b***)–(***e***) of figure 7.3, we see that the fattening artefacts appearing in the sparse disparity map $\mathfrak{D}^{(S)}$ and in $\mathfrak{D}_q^{(SF)}$ are replaced by far more plausible disparities in any of the displayed $\mathfrak{D}_q^{(\mathcal{P})}$ functions. This observation gave rise to a significantly more efficient algorithm

devised to filter these fattening artefacts from the sparse disparity maps.

7.3.1 Application: fattening effect removal revisited

The revised algorithm for the removal of fattening artefacts comprises two steps. First, we locate the points $(x, y) \in X$ having a disparity measure $\mathfrak{D}^{(S)}[x, y]$, close to the one of $\mathfrak{D}_q^{(\mathcal{P})}[x, y]$ for at least one $q \in \{10, 20, 30\}$. Such points are registered in a binary image $\mathbf{M}$ which, in order to summarise, is defined as:

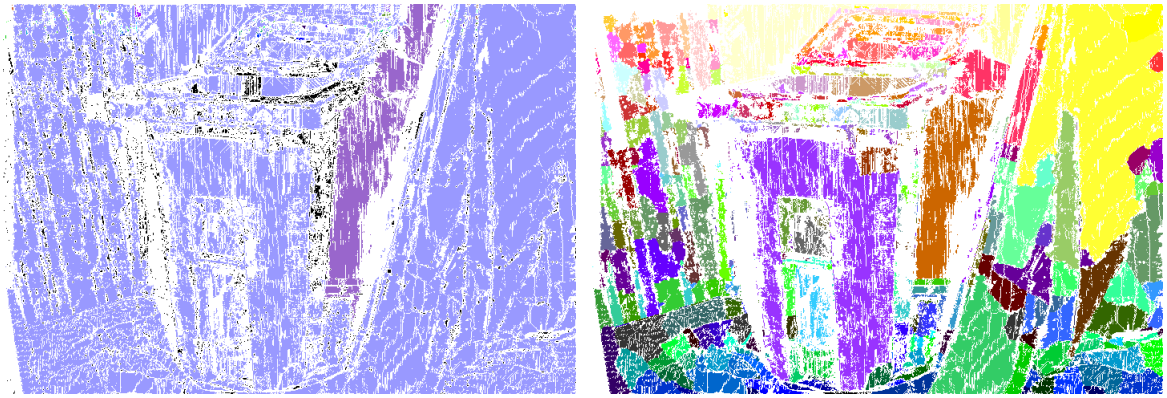
$$\mathbf{M}[x, y] = 1 \Leftrightarrow (x, y) \in X \text{ and } \exists\, q \in \{10, 20, 30\} \text{ s.t. } \left|\mathfrak{D}^{(S)} - \mathfrak{D}_q^{(\mathcal{P})}\right| [x, y] \leqslant \tau$$

τ, being the maximum difference in disparity allowed between the compared disparity maps, is set to a few pixels. Increasing this parameter will increase the span of the binary mask $\mathbf{M}$ across tilted regions, but will also increase the risk of appending pixels with bad disparity measures to the mask. Therefore, τ must remain small, independently of the scene configuration. In order to increase the span of the mask across tilted regions, we will use the binary mask $\mathbf{M}$ as a *marker* to *reconstruct* a special kind of disparity clusters. This action constitutes the second step of the algorithm.

The clustering of sparse disparity functions has been solved by algorithm 5.2. At a global scale, the fattening artefacts can belong to the same cluster as those points allocated correct disparity measures. This is the case in the `Recycle` instance, where one huge disparity cluster covers almost all the disparity measures, as can be observed in figure 7.4. Therefore, if all the measures of this cluster were reconstructed, the action of the filtering would be virtually nullified. At a regional scale though, we are able to attribute fattening artefacts more appropriately to separate clusters, which are not marked by $\mathbf{M}$. Therefore, instead of applying algorithm 5.2 to the entire domain of the disparity map $\mathfrak{D}^{(S)}$, we need to apply it separately to each cell of the related image partition $\mathcal{L}$, and then concatenate the produced clusters into a new cluster map $\mathcal{C}_\mathcal{L}$, so that no label appears simultaneously in different cells of $\mathcal{L}$. Let $\mathbf{M}^+$ be the binary image containing the points which remain after the filtering of the disparity map. $\mathbf{M}^+$ is then defined by equation 7.14 as :

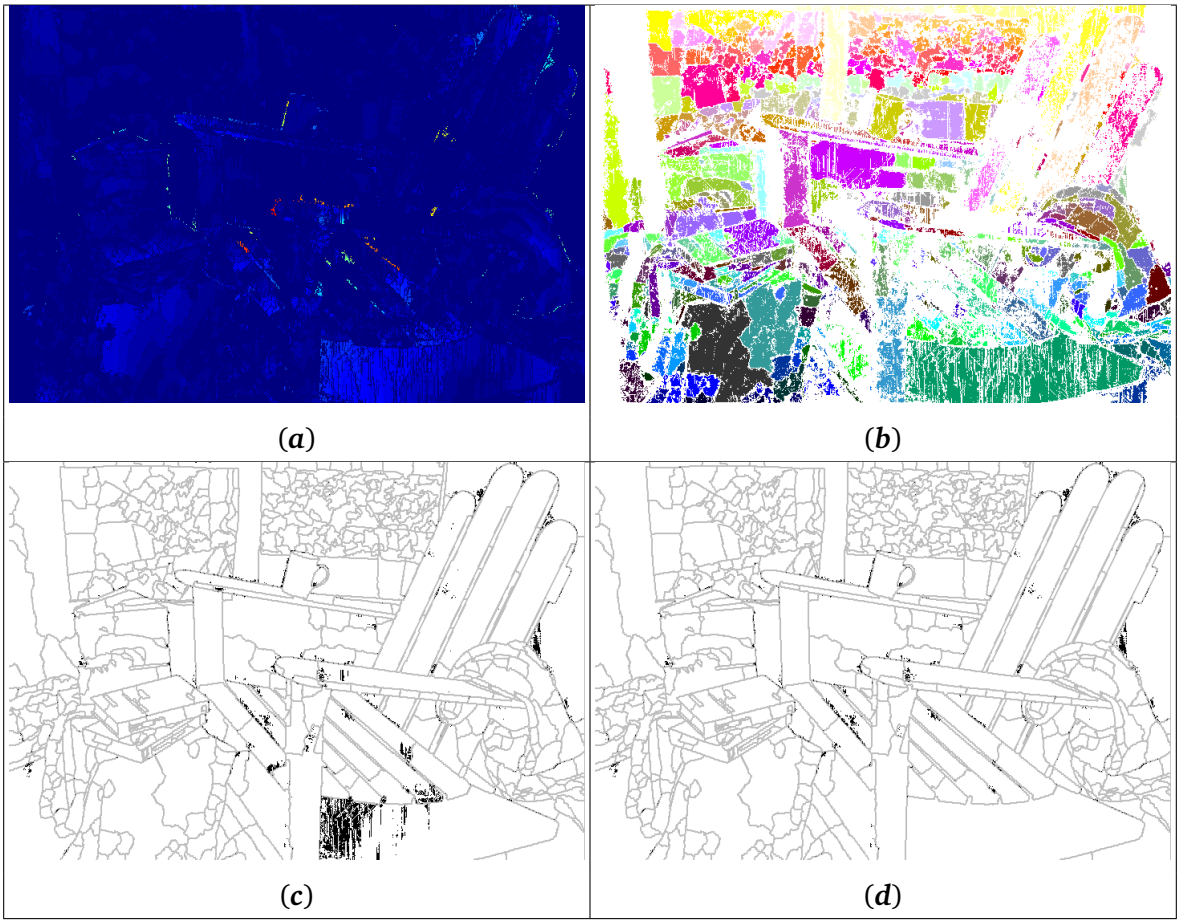
$$\mathbf{M}^+[x, y] = 1 \Leftrightarrow \exists\, (x', y') \text{ s.t. } \mathcal{C}_\mathcal{L}[x', y'] = \mathcal{C}_\mathcal{L}[x, y] \text{ and } \mathbf{M}[x', y'] = 1 \tag{7.14}$$

One final remark regarding the generation of the binary image $\mathbf{M}$: although the regional disparity map $\mathfrak{D}_q^{(\mathcal{P})}$ proved very useful in detecting the vast majority of fattening artefacts, those which constitute the sole disparity measures contained in the region to which they belong, could not be recognised by the method proposed thus far. In these particular cases, however, the occupancy of fattening artefacts remains a small percentage of the entire region area. It is



(a) (b)

FIGURE 7.4: Disparity clusters defined (a) across the whole image domain and (b) on a regional basis, for the `Recycle` instance.



(a) (b)

(c) (d)

FIGURE 7.5: Fattening effect removal on `Adirondack`. (a) The visualisation of $|\mathfrak{D}_q^{(SF)} - \mathfrak{D}_q^{(\mathcal{P})}|$ for $q = 20$ accentuates the detection of fattening artefacts. (b) The cluster map of disparities computed on a regional basis, $\mathcal{C}_{\mathcal{L}}$. ($c$) The binary image $\mathbf{M}$ computed for $\tau = 5$ pixels (points without a disparity measure appear in white, points with a disparity measure but enclosed in a region where measures cover less than 5% of the whole region area are deactivated) and (d) the binary image $\mathbf{M}^+$, superimposed on the segmentation.

therefore good practice to clear the regions containing insufficient disparity information of all their measures and to reflect this pruning by updating image $\mathcal{M}$ accordingly.

Figure 7.5 shows the detection of fattening artefacts on the `Adirondack` example. Furthermore, it is interesting to study the impact of updating the filtering blocks used within the multi-scale diffusion scheme introduced in section 5.2.3. In figure 7.6, we see, by changing the former fattening removal method for the new one, that further errors are avoided and that the disparity maps thus reach a higher level of accuracy. That confirms that the filtering stage in multi-scale approaches should never be neglected and that a simple mechanism of disparity map estimation, such as that presented in this section, can help to fulfil the requirements of a good filtering operator for sparse disparity maps.

7.4 Interpolation using distance functions on partitions

The objective of the two-pass multi-scale diffusion presented in section 5.2.3, is to enable the measurement of disparities across all the non-occluded image areas. The characteristic feature of this refinement is its ability to take account of the superimposition costs found in the disparity space volume. We are now about to fill the holes of the sparse disparity maps. Essentially, these holes cover occluded areas, for which the superimposition costs are meaningless. Therefore, we would expect the superimposition costs to have hardly any beneficial effect on the interpolation process. For that reason, we propose, in the first instance, to resort to the interpolation mechanism based on binary distance functions. In section 3.4, we presented an interpolation example for a one-dimensional function; here we have to extend it so that it works with a two-dimensional disparity function.

Distance functions of binary volumes Let $\mathfrak{D}^{(S)}$ be the sparse disparity map considered as input of the interpolator. The available disparity measures may be projected into a 3D relief $\mathfrak{D}_0$ expressed as follows:

$$\mathfrak{D}_0[x,y,d] = \begin{cases} 0 & \text{if } \mathfrak{D}^{(S)}[x,y] = d \\ +\infty & \text{otherwise} \end{cases}$$

The reader might have noticed that $\mathfrak{D}_0$ constitutes the starting point of the computation of a distance function. Using recurrence relation 3.14 with $\eta = 1$, we obtain the distance function of the binary mask $\mathcal{M}$ described by $S(\mathcal{M}) = \{(x,y,d) \mid \mathfrak{D}_0[x,y,d] = +\infty\}$. We shall refer to this distance function as $\mathfrak{D}$. If we invert $\mathfrak{D}$, we may compute its 3D watershed from the background and foreground markers covering initially the lowest and the highest disparity planes, respectively. The 3D watershed then constitutes a hyperplane enclosed in the volume $\mathfrak{D}$, and separating it into foreground and background voxels. Projecting onto each pixel of the image plane, the disparity at which this watershed occurs, results in a full disparity map such as the one displayed in figure

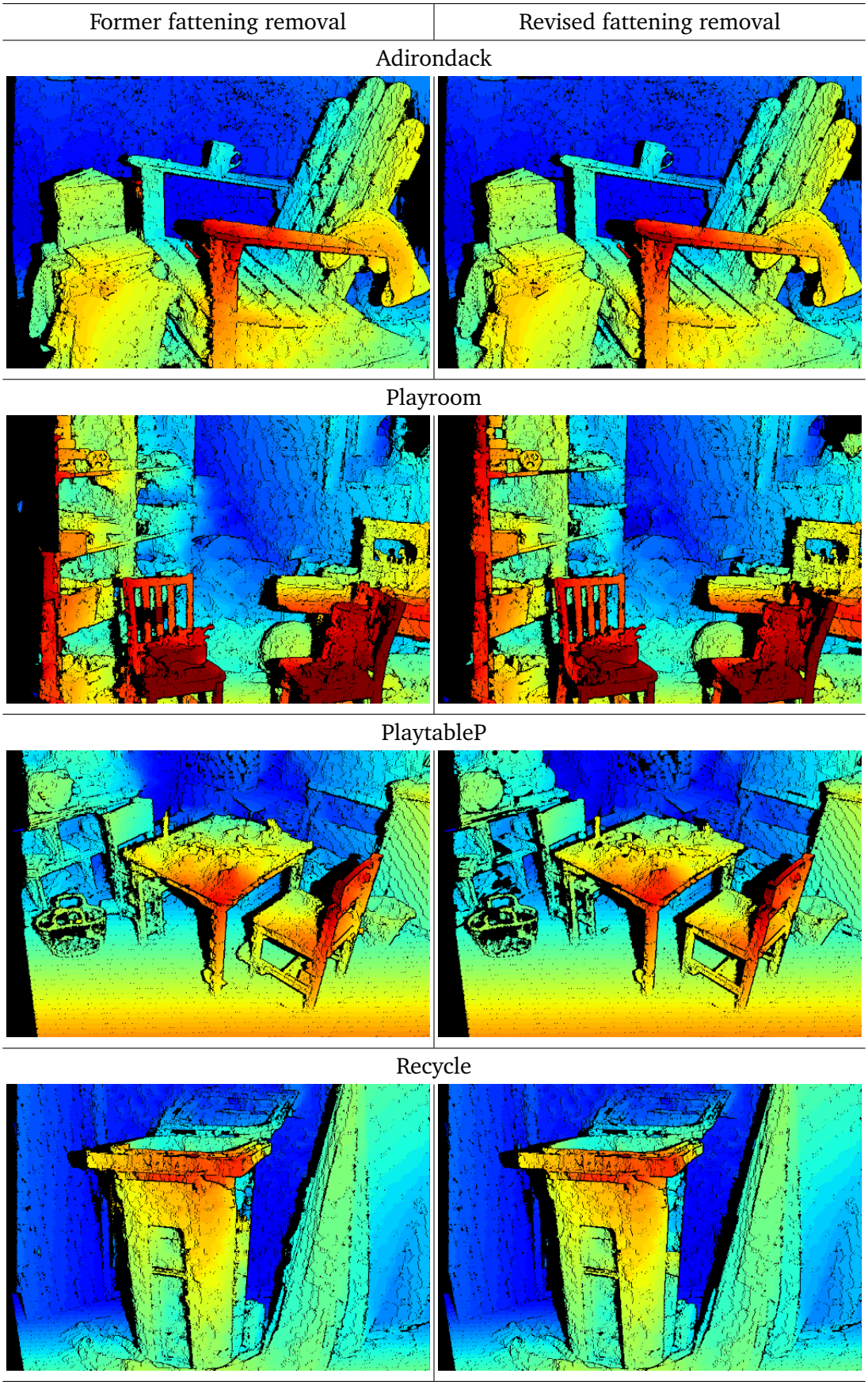
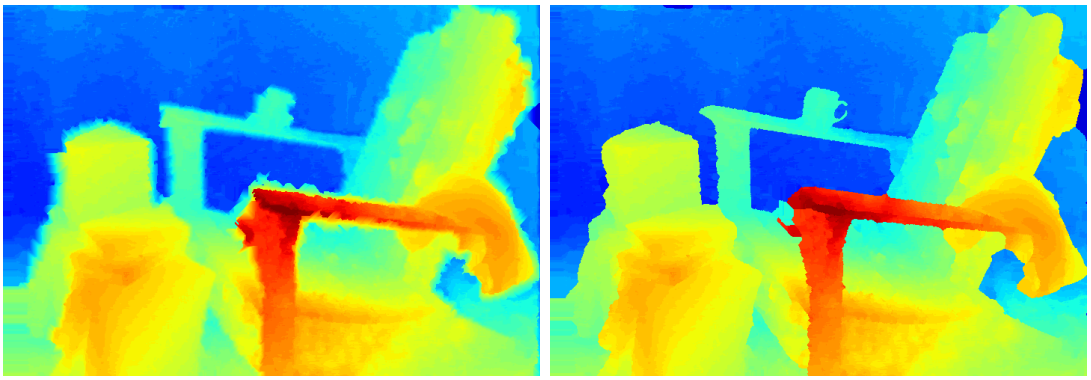


FIGURE 7.6: Comparison of the two proposed fattening removal methods. The disparity maps displayed are those obtained by the multi-scale diffusion mechanism, using the coarse-to-fine refinement, followed by the fine-to-coarse densification.



(a) (b)

FIGURE 7.7: Disparity maps deduced from the distance function $\mathfrak{D}$, given as input the sparse disparity map shown in figure 7.6 for the `Adirondack` test case. (a) Exact interpolation supllied by the 3D watershed of the inverted distance function, using the settings described in section 3.4. (b) Interpolation resulting from minimising equation 7.15.

7.7(a). We notice that the disparities surrounding the contours are blurred which, in hindsight, is not surprising. The problem derives from the fact that the distance function has been computed at a global scale. Therefore, across occluded areas, the disparities between occluded and occluding objects are simply interpolated.

Another way of using distance function $\mathfrak{D}$ to find the interpolated disparity map of $\mathfrak{D}^{(S)}$, consists of minimising equation 7.15, so that:

$$\mathfrak{D}[x, y] = \arg\min_{d} \mathfrak{D}[x, y, d] \tag{7.15}$$

For example, doing so results in the disparity map shown in figure 7.7(b). Concerning the areas of the image where the actual disparity function is continuous, we notice little difference between this map and that resulting from the interpolation based on the 3D watershed. The fattening artefact though replaces the blurring artefact, for the same reasons we mentioned earlier. Once more, for distance functions to be meaningful with respect to our objective of interpolating disparity maps, their computation must be performed on a regional basis. Later, we shall consider the interpolation based only on the minimisation of equation 7.15.

Distance functions on partition Let $\mathcal{L}$ be the partition of the reference image. P denotes the set of points lying on the borders of any cell in $\mathcal{L}$, such that, for any directional structuring element $\mathbf{h}$:

$$P = \{(x, y) \mid \exists\, \mathbf{h} \text{ s.t. } (\delta_{\mathbf{h}}(\mathcal{L}) - \varepsilon_{\mathbf{h}}(\mathcal{L}))[x, y] > 0\}$$

Furthermore, we call $\mathbf{M}^{(P)}$ the volume defined such that $\mathbf{M}^{(P)}[x, y, d] = +\infty$ if $(x, y) \in P$, otherwise $\mathbf{M}^{(P)}[x, y, d] = 0$. Let $\mathfrak{D}^{(P)}$ be a distance function computed using the recurrence

relation as:

$$\begin{aligned} \mathcal{D}_0^{(P)} &= \max\left\{\mathcal{M}^{(P)}, \mathcal{D}_0\right\} \\ \mathcal{D}_t^{(P)} &= \max\left\{\mathcal{M}^{(P)}, \min\left\{\mathcal{D}_{t-1}^{(P)}, \varepsilon_H\left(\mathcal{D}_{t-1}^{(P)}\right)+1\right\}\right\} \\ \mathcal{D}^{(P)} = \mathcal{D}_{t^\star}^{(P)} &\Leftrightarrow \mathcal{D}_{t^\star}^{(P)} = \mathcal{D}_{t^\star+1}^{(P)} \end{aligned} \tag{7.16}$$

$\mathcal{D}^{(P)}$ corresponds to the distance function of the binary volume $\mathcal{M}$, computed independently for the interior of each of the cells of partition $\mathcal{L}$. Substituting $\mathcal{D}$ with $\mathcal{D}^{(P)}$ in equation 7.15, we obtain a new interpolated disparity map, such as that illustrated in figure 7.8(*a*). Yet, the result is not perfect for two reasons. Firstly, due to the way $\mathcal{D}^{(P)}$ is constructed, it is impossible to allocate a disparity value along the contours of the partition cells. Secondly, no interpolation is possible across cells devoid of disparity measures in their interior. Nonetheless, we have only presented the core of the interpolation process. Section 7.4.1 concentrates exclusively on the intricacies of the final depth map estimation and provides its full algorithm.

7.4.1 Application: final disparity maps computation

The algorithm transforming the sparse disparity maps into final disparity maps using the aforementioned distance functions, requires as input:

- $\mathbf{I}$, the gradient magnitude of the image associated with the sought disparity map.
- $\mathcal{L}$, the partition of the image for which the disparity map is computed.
- $\mathfrak{D}^{(S)}$, the sparse disparity map resulting from the multi-scale diffusion presented in section 5.2.3, using the two-pass refinement scheme and the fattening effect removal described in section 7.3.1.

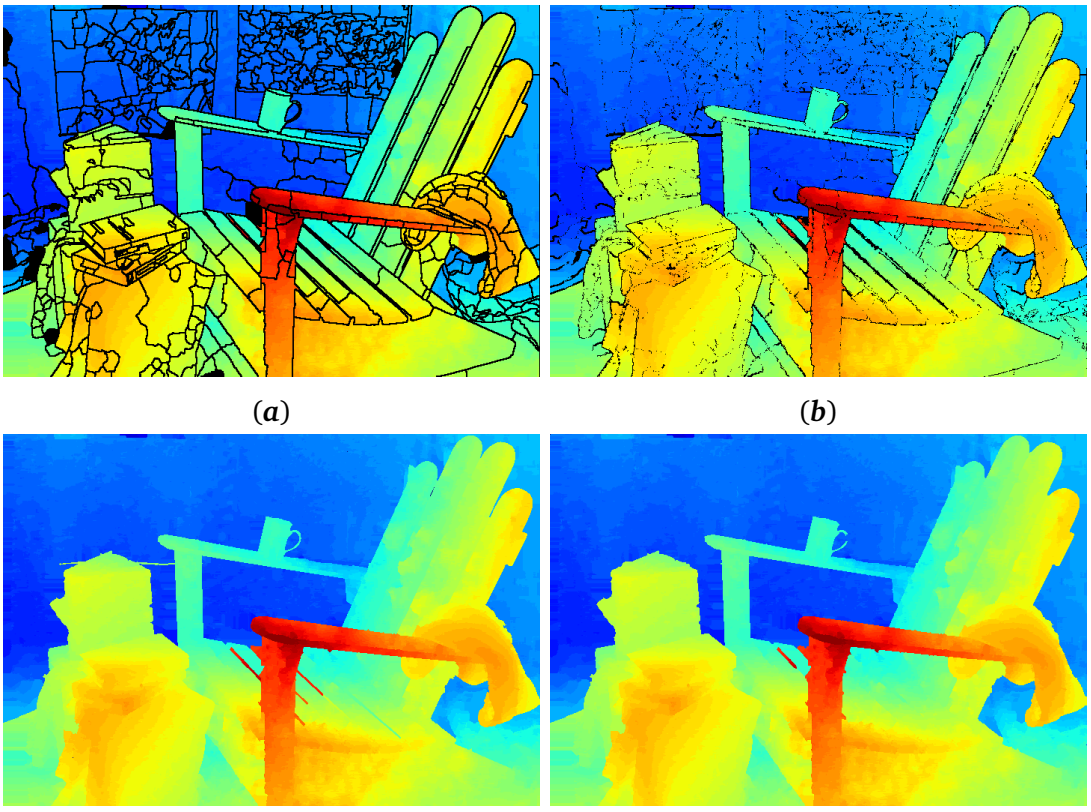
The algorithm then consists of the following high-level execution steps:

1. Locate cells of $\mathcal{L}$ which do not have a disparity measure in their interior. Mark these cells as “invalid”
2. Update $\mathcal{L}$, so that each invalid cell is assigned the label of a valid cell lying in its vicinity.
3. Compute the set P of points lying on the new cell borders.
4. Compute the distance function $\mathcal{D}^{(P)}$ according to equation 7.16.
5. Deduce the interpolated disparity map according to equation 7.15 and invalidate disparities for all points belonging to set P.
6. If available, restore the disparity measures from $\mathfrak{D}^{(S)}$ for all points belonging to set P.
7. Fill the remaining invalid points with a disparity lying in their vicinity, giving preference to points marked by same label in $\mathcal{L}$.

Steps 2 and 7 are certainly the most abstract in this description, and we shall now provide details necessary for their implementation.

Attributing new labels to invalidated cells There are two obstacles to overcome with this problem of relabelling. The first consists of merging invalid cells to valid cells. This fusion should remain meaningful; so that the likelihood of merged cells effectively segmenting the objects of the scene continues. Second, there is the potential for an invalid cell to be surrounded entirely by other invalid cells. In order to initiate the relabelling process, we compute the watershed transformation of the topographical surface represented by image $\mathbf{I}$ using $\mathcal{L}_0$ as the controlling map of markers. $\mathcal{L}_0$ is defined such that $\mathcal{L}_0[x, y] = 0$ if (x, y) belongs to an invalidated cell, otherwise $\mathcal{L}_0[x, y] = \mathcal{L}[x, y]$. Let $\mathcal{L}_\star$ be the output of this watershed transformation. We now have the guarantee that partition $\mathcal{L}_\star$ is fully covered by labels allocated to valid cells in $\mathcal{L}$. Using the gradient magnitude $\mathbf{I}$ to control the watershed allows two neighbouring cells sharing a frontier with little accentuation to be more easily merged. But supposing a cell becomes separated from its neighbours by frontiers of strictly identical gradient magnitudes all along its border, then this cell would be flooded by each of the surrounding lakes simultaneously and thus, would be covered with different labels. In order to take the final decision, an invalid cell of $\mathcal{L}$ is updated with the label in $\mathcal{L}_\star$ that covers the majority of its surface.

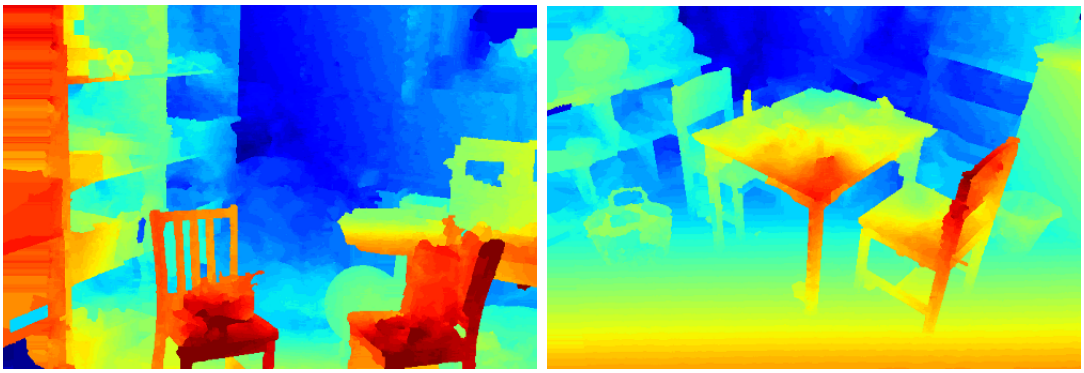
Final disparity map filling along the cell borders Algorithm 7.1 provides all the details about how to update and finalise the disparity map obtained at step 6. The missing disparities all lie along the contours of the cells composing partition $\mathcal{L}$. The idea of the filling mechanism is to assign the remaining invalid points to the smallest disparity found in their neighbourhood and *preferably* in the same region. We acknowledge that the adverb "preferably" might strike the reader as illogical: of course, if the regions denote objects in the scene, the disparity allocated to the invalid points must originate from the same region. However, experience showed that the fusion of thin and elongated structures with valid cells performed at step 2, is not systematically appropriate. Therefore, for these special structures, imposing the condition that the disparity originate from the same region, typically results in a higher number of critical errors compared to the scenario where only the regional membership preference is applied, in the direct neighbourhood of the image plane. This is illustrated by the `Adirondack` examples, displayed in figures 7.8(***c***) and 7.8(***d***).



(*a*) (*b*)

(*c*) (*d*)

FIGURE 7.8: Disparity maps resulting from the distance functions computed for each cell of the image partition. (*a*) Disparity map resulting from the minimisation of equation 7.15 for $\mathfrak{D}^{(P)}$. Image areas shown in black are the cell borders and the cells which do not have a disparity measure in their interior. (*b*) Disparity map obtained after step 6 of our estimation algorithm. (*c*) Final disparity map resulting from algorithm 7.1, replacing line 11 with `sl` $\leftarrow$ `True`, that is enforcing regional consistency for the filling of disparities. (*d*) Final disparity map resulting from algorithm 7.1.



(*a*) (*b*)

FIGURE 7.9: Output of algorithm 7.1 for (*a*) `Playroom` and (*b*) `PlaytableP` test cases

Algorithm 7.1 Filling of contour disparities

```
1:  function FillContourDisparities(𝔇, ℒ)
2:      𝔇′ ← 𝔇
3:      𝔇 ← 𝔇 × 0

4:      Re-iterate until process converges
5:      while 𝔇′ ≠ 𝔇 do
6:          𝔇 ← 𝔇′
7:          for all (x, y) left without a disparity in 𝔇 do
8:              Disparity to be assigned to (x, y)
9:              d ← +∞
10:             Token indicating whether disparity stems from the same region
11:             sl ← False

12:             for all (x′, y′) lying in the neighbourhood of (x, y) do
13:                 if (x′, y′) has a disparity in 𝔇 then
14:                     if ℒ[x′, y′] = ℒ[x, y] then
15:                         Processing a disparity discovered in the same region
16:                         if sl = False then
17:                             Disparities discovered in the same region have the priority
18:                             sl ← True
19:                             d ← 𝔇[x′, y′]
20:                         else if d > 𝔇′[x′, y′] then
21:                             d ← 𝔇[x′, y′]
22:                     else if sl = False then
23:                         Processing a disparity discovered in a different region
24:                         if d > 𝔇[x′, y′] then
25:                             d ← 𝔇[x′, y′]

26:             Update disparity of current point
27:             𝔇′[x, y] ← d
```

Summary

The choice of an estimation method for the computation of disparity maps depends on the nature of the measured disparity data. This is predominantly characterised by its sparsity and by its reliability. In this chapter, we dealt with three types of measures: feature point disparities, sparse disparity maps produced by the multi-scale diffusion system, and regional disparities. In the first two cases, the data is assumed to be fully accurate and the purpose of the estimation method is to generate the disparities of the points for which no measure is available. In the third case, the regional disparities measured on the coarse and fine partitions of the image of interest, are interpreted as observations of the actual regional disparity function for a fine level of segmentation. The purpose of the estimation algorithm is to discover this regional disparity map in such a manner that it best fits the observations, while satisfying a plausible variational model. This model has remained virtually the same in all scenarios: the estimated disparity function, whether it takes its values on a pixel basis or on a regional basis, must evolve smoothly across the main regions of the partition accompanying the disparity map. To proceed, each of these main regions has been processed independently in conjunction with: a linear variogram for the kriging technique, adequate pairwise potentials for the MAP inference with pairwise MRFs, and distance functions to interpolate the holes of the sparse disparity maps.

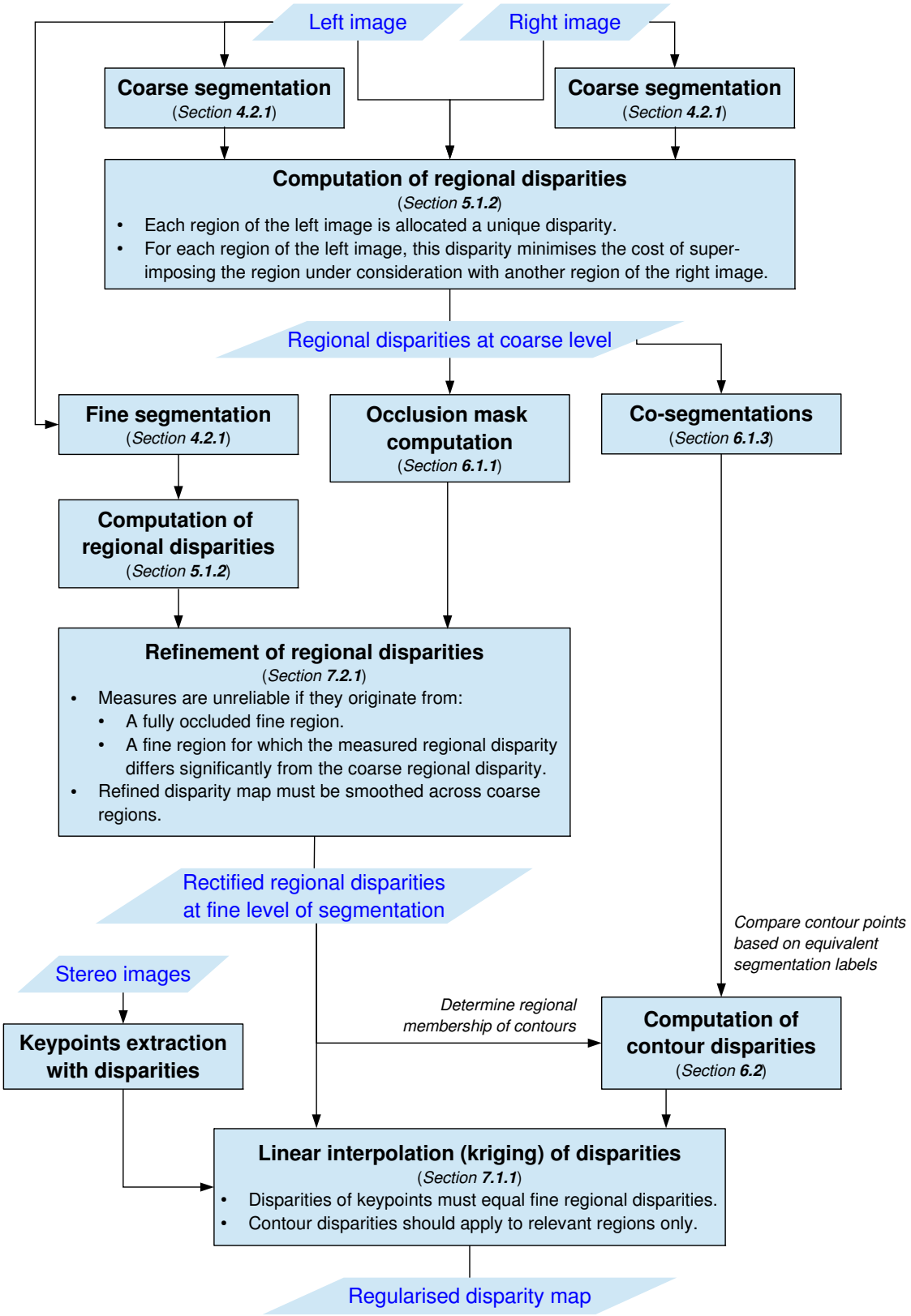
This summary concludes our methodology on the estimation of depth maps. We shall now provide the reader with an analysis of the results obtained on the Middlebury 2014 benchmark. Perhaps the main limitation of the methods proposed throughout this study, is the binary way in which segmentations control the measurement and the estimation algorithms, explaining most of the artefacts which occasionally remain in the final results. Some suggestions about how to improve this aspect of the methodology will be provided.

Appendix

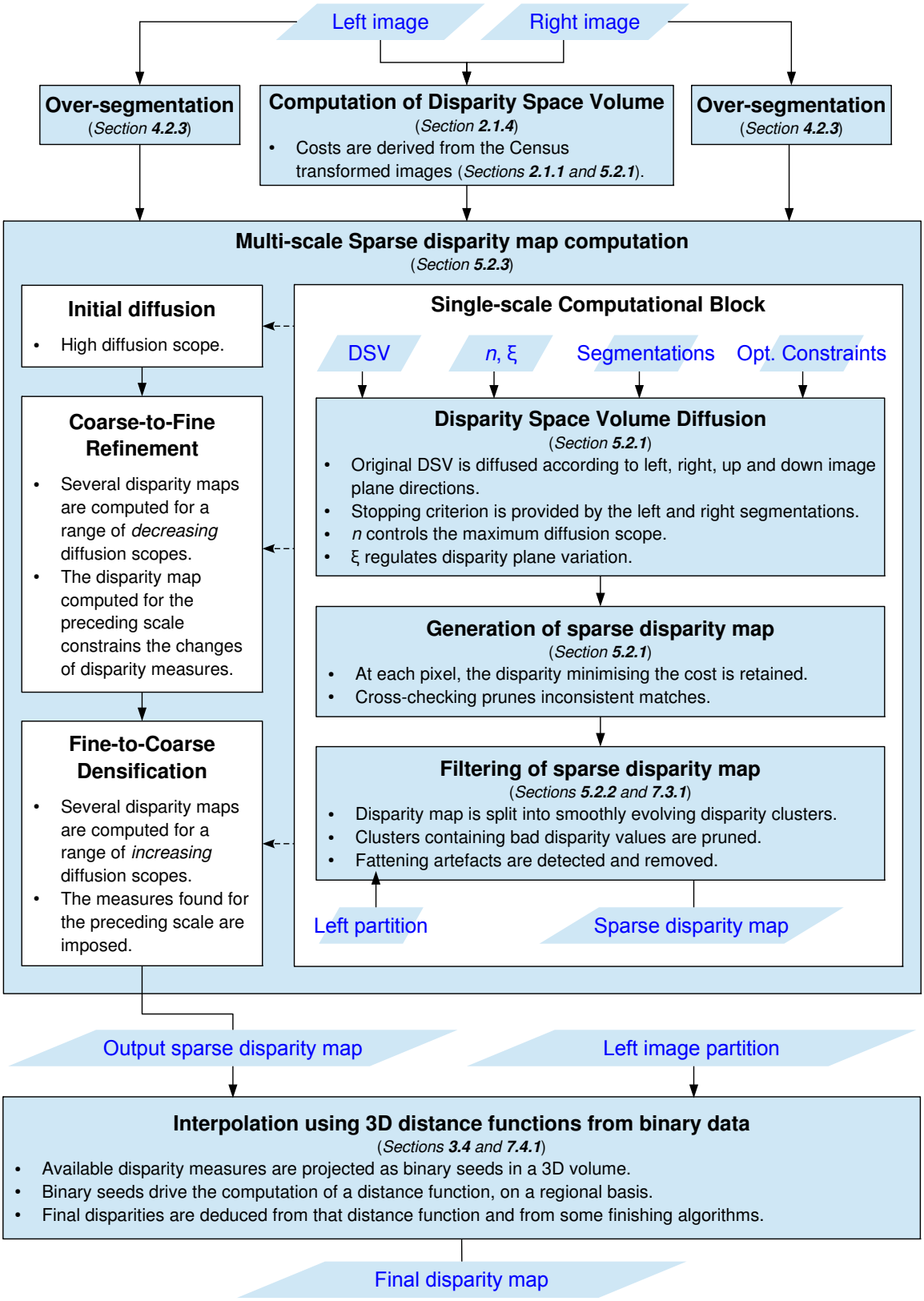
In this work, two major frameworks were developed for the computation of disparity maps from pairs of stereo images. The first is based on the utilisation and refinement of regional disparities. The second concerns the multi-scale aggregations of individual pixel superimposition costs.

The first part of this appendix provides the block diagrams corresponding to these two distinct frameworks. For the most important blocks of each diagram, the reader will be referred to the sections where implementation details can be found. The second part of this appendix is devoted to the experiments performed on the Middlebury 2014 dataset using the second framework: a quantitative assessment is performed and related to a more objective evaluation of the algorithm's features and qualities.

A.1 Framework 1: Multi-scale regional disparities



A.2 Framework 2: Multi-scale regional aggregations



A.3 Experiments conducted on the Middlebury 2014 dataset

Since September 2014, the Middlebury website [1] has provided a modern stereo vision dataset comprising high quality photographs of interior scenes. In contrast to the 2002 dataset, these scenes are composed of objects of various shapes, with or without texture. There is also a significant number of thin and tilted objects, requiring that stereo algorithms operate without the ordering constraint and without the fronto-parallel assumption. For that reason, the experiments performed on this dataset resort to our algorithm based on multi-scale diffusions, as illustrated by the flowchart of framework 2.

Exploited features

Resolution and baseline Three different kinds of image dimensions are provided for each image of the dataset. At full dimensions, images have a size in the region of 2900×2000 pixels. Due to the amount of memory required by our disparity space volumes, we opted for the quarter resolution, where image dimensions are close to 725×500 pixels. For such dimensions, the maximum disparity observed reaches 200 pixels.

Calibration data Each stereo pair is accompanied by a calibration file, containing the calibration parameters of the cameras, as well as a maximum disparity threshold. In the experiments presented in this appendix, this threshold was used to constrain the disparity search space considered by our stereo algorithms.

Rectification The stereo images do not have a perfect calibration, which implies that some vertical disparities exist between corresponding pixels. In the quarter resolution dataset though, most of the occasional vertical disparities are smaller than 1 pixel, and therefore we did not perform the rectification. This, of course, will severely affect our results on the `Playtable`, `Australia` and `Crusade` test cases, where the vertical disparities resulting from the non-rectified configuration are far more significant. Therefore, the output of our algorithm is only relevant to the rectified versions of these images, which are provided as `PlaytableP`, `AustraliaP` and `CrusadeP`.

Quality metrics

Accuracy measures for full disparity maps The Middlebury 2014 benchmark system compares the performances of several methods of disparity map estimation, by analysing the accuracy of the disparity maps obtained for the proposed dataset. To proceed, each disparity map is

[1] Please, visit the Middlebury website at http://vision.middlebury.edu/stereo/

compared to a ground truth, obtained at the shooting stage using the structured lighting technique described in [Scharstein et al., 2014]. The principal metrics considered for an accuracy assessment of disparity maps are:

- The ratio of **bad pixels** – This measure indicates the percentage of pixels which are allocated a disparity measure which differs from the ground truth, by a magnitude equal to or greater than 1 pixel, at quarter resolution.
- **Average error** – This measure indicates the mean average of the absolute difference between the computed and ground truth disparities. We express this error in terms of pixels, with respect to the quarter resolution images.
- **RMS error** – This measure is the root-mean-square error observed between the computed and ground truth disparities. By nature, the RMS error is more sensitive to high disparity discrepancies than is the average error. In this evaluation, the RMS error is also expressed in terms of pixels, with respect to the quarter resolution images.

Accuracy measures for sparse disparity maps The analysis of sparse disparity maps is slightly more complex than that of full disparity maps, since we have to deal with pixels having no disparity measure. The three metrics mentioned for the full disparity maps will also be considered within the analysis of sparse disparity maps, but we need to consider the following:

- **Bad pixels** – The ratio of bad pixels is expressed according to the full image area. Pixels having no disparity measure are considered neither "good" nor "bad", but belong to a third class, which we will call "invalid". This choice was made in order to remain consistent with the "Bad Pixels 4.0" metric proposed by the online benchmarking system.
- **Average error** – Invalid pixels are not taken into account.
- **RMS error** – Invalid pixels are not taken into account.

In addition to these three standard metrics, we provide:

- The **precision** of the disparity map – This measure corresponds to the proportion of "good" pixels relative to all the pixels of the image plane which are allocated a disparity measure.

Coverage measures for sparse disparity maps An important aspect of the evaluation of sparse disparity maps is the image plane coverage of disparity measures, which gives an indication of the density of the disparity map. We provide two different types of measure:

- **Occupancy ratio** – This measure provides, with respect to the full image plane, the proportion of pixels being allocated a disparity measure.
- **Occluded ratio** – This measure indicates, with respect to the truly occluded image areas, the proportion of pixels which have been allocated a disparity measure. Ideally, disparity measures should *not* be assigned to occluded areas, thus we expect low results for that particular metric.

Filtering quality The filtering stage of sparse disparity map computation is quite specific to this work, and therefore deserves some attention within the evaluation. We already provide the precision measure for all sparse disparity maps. When comparing the filtered results with the initial results, we shall also provide the filter **recall**. This corresponds to the proportion of pixels in the initial disparity map, which have been preserved by the filter and have been allocated a "good" disparity measure.

Further considerations Since the handling of occlusions has been an important focus of this work, **all pixels** of the image domain, including those in occluded areas, participate in the accuracy measurements. When comparing the results presented in this appendix, to those available in the Middlebury benchmark, it is important to notice that every result presented in the benchmark relates to the full resolution images. Since we are working at quarter resolution, the bad pixel ratio corresponds to the "Bad Pixels 4.0" metric. Furthermore, the errors expressed in terms of pixels should be multiplied by a factor of four, in order to compare them with errors observed at full resolution.

Sparse disparity map quantitative evaluation

The results regarding the quality of our sparse disparity maps, have been assembled in tables A.1 and A.2. We show the evolution of the quality measures after different stages of our multi-scale computational system: we consider the sparse disparity maps obtained immediately after the initial cost diffusion, those obtained after the filtering step, those obtained after the multi-scale refinement step, and finally those resulting from the fine-to-coarse densification. For each metric, we provide the mean average for the 13 images presented at the top of the list. Due to the non-ideal calibration, our method still has difficulty handling the `Playtable` and `Shelves` test cases. Thus their associated disparity maps are not comparable to those obtained for other images of the dataset. Therefore, the quality measures obtained from these test cases do not contribute to the displayed mean averages.

Comments The evolution of the mean averages computed for each metric and for each stage of the disparity map computation process, shows that the filtered sparse disparity maps resulting from the single-scale diffusion are the most precise, with, at quarter resolution, 92.3% of pixels allocated a disparity measure corresponding to the ground truth. The mean error is a disparity of approximately half a pixel, whilst, at the time of writing, the RMS error ranks these sparse disparity maps third, amongst the 44 methods displayed in the benchmark. The issue with these filtered disparity maps, is the occupancy of the image plane, which is slightly higher than 45% of the full image area. The refinement of the disparity maps allows the recovery of a level of densification close to 60%, similar to that observed for the initial disparity maps, whilst it produces disparity measures with a level of accuracy comparable to that obtained for filtered

Initial sparse disparity maps (single, high level diffusion)							
Image name	*Bad pixel* ratio	*Mean error* in pixels	*RMS error* in pixels	*Occluded* ratio	*Precision* ratio		*Occupancy* ratio
Adirondack	0.077	1.600	6.019	0.103	0.859		0.547
ArtL	0.079	1.833	5.736	0.109	0.862		0.573
Jadeplant	0.080	4.930	16.470	0.101	0.834		0.483
Motorcycle	0.105	1.762	6.045	0.128	0.820		0.583
MotorcycleE	0.096	1.630	5.798	0.125	0.835		0.582
Piano	0.087	1.331	4.268	0.180	0.863		0.633
PianoL	0.097	2.049	6.073	0.153	0.808		0.506
Pipes	0.077	2.022	6.709	0.140	0.868		0.585
Playroom	0.113	2.096	6.726	0.113	0.782		0.518
PlaytableP	0.099	1.305	4.447	0.148	0.836		0.605
Recycle	0.075	1.185	4.445	0.097	0.881		0.631
Teddy	0.051	1.163	4.505	0.094	0.921		0.646
Vintage	0.093	3.196	10.565	0.093	0.755		0.379
Mean average	**0.087**	**2.008**	**6.754**	**0.122**	**0.840**		**0.559**
Playtable	0.253	4.743	8.912	0.185	0.525		0.533
Shelves	0.174	2.592	5.577	0.181	0.659		0.511
Filtered sparse disparity maps (single, high level diffusion)							
Image name	*Bad pixel* ratio	*Mean error* in pixels	*RMS error* in pixels	*Occluded* ratio	*Precision* ratio	*Filter recall* ratio	*Occupancy* ratio
Adirondack	0.035	0.488	1.469	0.029	0.925	0.915	0.465
ArtL	0.022	0.517	1.822	0.026	0.952	0.891	0.462
Jadeplant	0.011	0.631	4.465	0.013	0.971	0.903	0.375
Motorcycle	0.054	0.566	1.788	0.041	0.889	0.902	0.485
MotorcycleE	0.051	0.544	1.810	0.044	0.896	0.907	0.492
Piano	0.032	0.493	1.307	0.087	0.940	0.921	0.535
PianoL	0.033	0.528	1.224	0.061	0.916	0.885	0.395
Pipes	0.021	0.639	2.603	0.053	0.954	0.860	0.458
Playroom	0.054	0.644	1.807	0.037	0.870	0.894	0.416
PlaytableP	0.049	0.504	1.125	0.043	0.904	0.907	0.508
Recycle	0.035	0.426	0.978	0.033	0.938	0.948	0.562
Teddy	0.013	0.379	1.303	0.036	0.977	0.911	0.555
Vintage	0.037	0.848	2.927	0.024	0.872	0.881	0.289
Mean average	**0.034**	**0.554**	**1.894**	**0.041**	**0.923**	**0.902**	**0.461**
Playtable	0.162	3.438	5.889	0.072	0.595	0.850	0.400
Shelves	0.093	1.407	3.309	0.106	0.765	0.899	0.396
Goal	Minimise	Minimise	Minimise	Minimise	Maximise	Maximise	Maximise

TABLE A.1: Quantitative performances on the Middlebury 2014 training set

Sparse disparity maps after the multi-scale coarse-to-fine refinement and filtering						
Image name	*Bad pixel* ratio	*Mean error* in pixels	*RMS error* in pixels	*Occluded* ratio	*Precision* ratio	*Occupancy* ratio
Adirondack	0.055	0.513	1.478	0.043	0.911	0.621
ArtL	0.033	0.623	2.083	0.040	0.937	0.526
Jadeplant	0.031	0.971	5.299	0.019	0.926	0.419
Motorcycle	0.032	0.554	2.191	0.063	0.948	0.620
MotorcycleE	0.026	0.492	1.975	0.059	0.959	0.629
Piano	0.065	0.650	1.770	0.117	0.900	0.653
PianoL	0.074	0.813	2.068	0.094	0.865	0.548
Pipes	0.031	0.697	2.770	0.061	0.941	0.527
Playroom	0.090	0.779	2.139	0.052	0.838	0.556
PlaytableP	0.067	0.549	1.243	0.067	0.900	0.672
Recycle	0.060	0.532	1.344	0.042	0.911	0.674
Teddy	0.023	0.454	1.876	0.051	0.965	0.652
Vintage	0.059	1.149	3.651	0.037	0.855	0.408
Mean average	**0.050**	**0.675**	**2.299**	**0.057**	**0.912**	**0.577**
Playtable	0.299	3.645	5.778	0.131	0.526	0.631
Shelves	0.258	2.627	5.086	0.214	0.592	0.633
Sparse disparity maps after the multi-scale fine-to-coarse densification and filtering						
Image name	*Bad pixel* ratio	*Mean error* in pixels	*RMS error* in pixels	*Occluded* ratio	*Precision* ratio	*Occupancy* ratio
Adirondack	0.087	0.577	1.529	0.081	0.892	0.804
ArtL	0.055	0.844	2.693	0.074	0.911	0.616
Jadeplant	0.074	1.945	7.773	0.051	0.873	0.584
Motorcycle	0.049	0.703	2.813	0.124	0.934	0.747
MotorcycleE	0.040	0.622	2.605	0.120	0.947	0.757
Piano	0.113	0.865	2.246	0.192	0.858	0.794
PianoL	0.116	0.984	2.517	0.158	0.935	0.705
Pipes	0.061	1.093	3.894	0.144	0.907	0.653
Playroom	0.132	0.888	2.406	0.086	0.815	0.714
PlaytableP	0.092	0.618	1.412	0.107	0.883	0.783
Recycle	0.086	0.624	1.679	0.067	0.897	0.833
Teddy	0.041	0.563	2.471	0.089	0.947	0.780
Vintage	0.139	2.305	5.961	0.080	0.777	0.624
Mean average	**0.083**	**0.972**	**3.007**	**0.106**	**0.883**	**0.723**
Playtable	0.356	3.447	5.617	0.187	0.541	0.776
Shelves	0.357	2.953	5.514	0.301	0.553	0.798
Goal	Minimise	Minimise	Minimise	Minimise	Maximise	Maximise

TABLE A.2: Quantitative performances on the Middlebury 2014 training set (continued)

Full disparity maps			
Image name	*Bad pixel* ratio	*Mean error* in pixels	*RMS error* in pixels
Adirondack	0.148	0.813	2.411
ArtL	0.211	1.800	4.541
Jadeplant	0.320	6.040	15.740
Motorcycle	0.142	1.693	5.724
MotorcycleE	0.127	1.612	5.592
Piano	0.213	1.253	2.960
PianoL	0.297	2.072	4.845
Pipes	0.252	3.416	8.094
Playroom	0.299	1.928	5.295
PlaytableP	0.165	0.910	2.338
Recycle	0.151	0.946	2.511
Teddy	0.134	0.992	3.679
Vintage	0.384	4.363	9.106
Mean average	**0.219**	**2.141**	**5.603**
Playtable	0.477	3.576	5.940
Shelves	0.467	3.159	5.828
Goal	Minimise	Minimise	Minimise

TABLE A.3: Quantitative performances on the Middlebury 2014 training set (continued)

disparity maps with a single diffusion. The multi-scale densification on average allocates disparity values to 72.3% of the full image plane. However, we observe a decrease in the proportion of pixels with a disparity measure matching that of the ground truth with an error tolerance of 1 pixel. Nonetheless, the mean and RMS errors remain relatively small, which explains why the disparity maps produced remain perceptually appealing. In terms of RMS error, these densified disparity maps attain eighth position in the benchmark.

Full disparity map quantitative evaluation

The results regarding the quality of the full disparity maps resulting from our interpolation method, are assembled in table A.3. About a fifth of the disparity values differ from the ground truth, by a magnitude of 1 pixel or more, which is a borderline performance compared to competing methods, since our algorithm is ranked 28th amongst the 44 registered methods. However, the mean error remains relatively small and the ranking according to the RMS error allows our method to once more gain the 8th position with respect to the online benchmark. This is not surprising, given the perceptual quality of our final disparity maps.

Conclusion to the quantitative analysis

As a conclusion to this quantitative analysis, we would ascertain that, compared to current approaches to the computation of full disparity maps, our method performs well with respect to the minimisation of the RMS disparity error. This can be explained by the fact that the removal of fattening artefacts has been one of the focuses of this work, and that sharp discontinuities of disparity functions are ensured by the use of our over-segmentations within the aggregation and the interpolation phases. Finally, the pixels allocated the most accurate disparity measures can be easily recovered from our sparse disparity maps.

Conclusion

Despite more than forty years of research on the topic, the computation of depth maps from a pair of stereo images still constitutes a challenging task, as demonstrated by recent stereo databases. The principal difficulties arise from the considerable amount of homogeneous regions appearing in the stereo images, the unpredictability of their actual 3D shape in the scene, the denial of geometrical assumptions such as the ordering constraint, and the abundance of occluded areas.

In this work, we have proposed two major frameworks enabling the computation of depth maps according to distinct scenarios. The first framework uses the concept of regional disparities, whereby pixels belonging to the same region are matched in one block instead of being matched individually. This procedure yields visually appealing disparity maps, characterised by sharp and precise disparity discontinuities with no ambiguity across homogeneous regions. Whilst the method proved particularly useful in microstereopsis and on low-baseline stereo imagery, it relies on the fronto-parallel assumption which may be unrealistic in modern scenarios. For that reason, a second framework has been proposed. The latter is based on the diffusion of superimposition costs within a disparity space volume and is better suited to wide-baseline stereo imagery.

Common to both frameworks is the separation of the measurement phase from the estimation phase of the disparity map computation. A key feature of the measurement phase has been the handling of occlusion problems. In particular, we ensured that the superimposition costs of regions which have nothing in common, do not become mixed with the costs of regions which effectively match. Furthermore, we paid particular attention to the removal of fattening effects occurring across the areas of the stereo images which were both homogeneous and occluded.

The development of segmentation algorithms, which produce partitions relevant to the analysis of stereo superimpositions, has been essential in fulfilling these objectives. In the context of regional disparities, coarse and fine segmentations related by a hierarchical dependence have been generated, whereas a single fine partition has been used within the diffusion method. For this latter partition, we imposed the constraints that homogeneous regions should not be exaggeratedly over-segmented and that the minimum area of a region should depend on its saliency in the scene, so as to increase the chances of efficiently capturing thin and contrasted structures.

Mathematical morphology has of course played a pivotal role in the generation of the aforementioned segmentations. But it has also proved essential in the multi-scale analyses performed at different stages of this work. For example, the enhanced regularised gradient, which has been utilised to generate the coarse segmentations of defocused images, is fully driven by morphological operators. The same applies to the filtering component of the multi-scale cost diffusion system, requiring the clustering of disparities and the removal of fattening artefacts. Our study confirmed the importance of the filtering component intervening in multi-scale frameworks, and that it is key to obtaining good results. In the case of the enhanced regularised gradient, the filtering essentially resides in the levelling and the removal of fused contours. In multi-scale stereo analysis, the filtering consists of pruning bad matches.

The purpose of multi-scale analysis in stereo is to refine the disparity maps. In the case of cost diffusion, the transition from the coarse analysis to the fine one prevents the erroneous measures, which generally arise from correlation windows of insufficient size, whilst offering the high standard of accuracy typically found at fine scales. We conducted a similar refinement procedure from a coarse to a fine partition, when performing MAP inference. Finally, the fine-to-coarse diffusion employed within the diffusion scheme increases the density of the measured data.

MAP inference has been one of the estimation techniques studied in this work and is the only technique capable of correcting erroneous disparities while estimating a full disparity map. In the other techniques, we assumed that the disparity measures were filtered and accurate, and that the purpose of the estimation procedure was simply to fill in the holes of the sparse disparity maps. For those disparity maps obtained by using the multi-scale cost diffusion, we chose the interpolation based on cell distance functions, thereby producing results which are comparable to those generated by state-of-the-art methods. We also proposed an estimation approach based on linear kriging: for situations requiring regularisation of disparity functions, where disparities take floating values, this method may be extremely useful. Finally, for all these estimation techniques, we encouraged disparity functions to evolve smoothly across each of the main regions accompanying the image for which the disparity map is computed.

Perspectives

Before concluding this thesis, we propose some suggestions about how specific areas of this work could be improved, in order to avoid the imprecisions or errors which currently remain in our final disparity maps.

Non-blind estimation The estimation methods proposed in chapter 7 rely exclusively on the measured disparity data, without resorting to the disparity space volume. For that reason, we could have said that they are *blind* to the stereo image superimpositions. This is acceptable if

all the measures across non-occluded areas have been obtained and are pertinent, since the disparity space volume is not informative about the disparities of occluded areas. Besides, this was the point of separating the measurement phase from the estimation phase. However, for image areas left without disparity measures and not occluded in the other image of the stereo pair, the replacement of the regular distance by the generalised geodesic distance related to the disparity space volume, could improve the accuracy of the full disparity map. Likewise for the MAP inference, the unary terms could take into account the cost associated with each possible regional disparity.

Thorough exploitation of segmentation hierarchies Although mathematical morphology offers a wide range of hierarchical segmentations, these have only been touched upon in this work. When discussing the MAP inference, only two levels of segmentations have been used in order to provide a binary criterion stating whether or not two adjacent nodes in the fine partition should be linked. It is possible to imagine a more permissive scheme, with a single MRF modelling the entire regional disparity map at the finest level of the segmentation hierarchy. The strength of the pairwise term between two regions could for instance decrease as the resistance of the watershed separating these two regions increases with respect to the segmentation hierarchy. Furthermore, it could also be interesting to see how cost diffusions can be combined meaningfully for different levels of segmentation.

Towards 3D reconstruction The methods developed in this thesis have been presented on a modular basis, rather than on a framework basis. We opted for such a presentation, because many algorithms such as those of the enhanced gradient, the adaptive over-segmentation, the interpolation based on distance functions and kriging, may find other applications in image processing. Dedicated tools such as the fattening artefact removal and the correlation across intersections of the partition cells, may also be integrated to the pipeline of other depth estimation approaches. Finally, the framework based on the cost diffusion, produced results on wide-baseline imagery, which seem sufficiently accurate to infer depth data from multiple views, the combination of depth data paving the way to the reconstruction of 3D scenes.

Remerciements

Je ne saurais terminer cet ouvrage sans remercier tous ceux et celles qui m'ont apporté leur soutien durant ces trois années de préparation au doctorat.

En premier lieu, je tiens à exprimer ma profonde gratitude à mes deux directeurs de thèse : Serge Beucher et Michel Bilodeau. Tous deux m'ont formé et guidé pas à pas dans les méandres de la morphologie mathématique, dont je reconnais aujourd'hui l'importance et l'utilité dans la résolution des problèmes d'analyse d'images. Je leur témoigne toute ma reconnaissance pour leur disponibilité, leur suivi attentif et la confiance accordée durant le déroulement de cette thèse.

Durant mon doctorat, j'ai eu l'opportunité exceptionnelle de participer au projet de recherche européen PANORAMA, qui m'a permis de découvrir une facette du traitement d'images plus industrielle, notamment dans les secteurs de l'imagerie médicale, de la surveillance et de la télévision. J'aimerais en saluer tous les membres, en particulier Klaas Jan Damstra, chef de projets chez Grass Valley Pays-Bas, avec qui nous avons élaboré le démonstrateur de télévision 3D basé sur l'exploitation de vidéos stéréoscopiques, ainsi que Peter De With et François Bremond qui ont accepté de rejoindre mon jury de thèse en tant que rapporteurs.

Je tiens à saluer tout spécialement Sabine Süsstrunk. Ayant été son étudiant dans les cours de sciences de l'image, de la couleur et de la photographie digitale à l'EPFL, j'ai été très honoré qu'elle accepte de présider mon jury de thèse, et je l'en remercie de tout cœur.

En second lieu, je remercie chaleureusement tous les chercheurs permanents du Centre de Morphologie Mathématique, pour leurs conseils et leur attention. Les doctorants et post-doctorants que j'ai rencontrés ont rendu mon séjour à Fontainebleau d'autant plus plaisant. Je les remercie pour leur générosité et leur bonne humeur, et je les salue avec toute mon affection: Sébastien Drouyer, Pierre Guillou, Jean-Baptiste Gasnier, Vaïa Machairas, Haisheng Wang, Robin Alais, Théodore Chabardes, Enguerrand Couka, Amin Fehri, Gianni Franchi, Amira Belhedi, Dieu-Sang Ly, Albane Borocco, Antoine Goblet, Borja Millán, Adrián Colomer Granero, Joris Corvo, Andres Serna-Morales, Bassam Abdallah et Emmanuel Chevallier. J'adresse par ailleurs mes pensées les plus sincères à Anne-Marie De Castro et Catherine Moysan qui s'occupent si bien de nous et du centre.

Je renouvelle ma tendresse à mes parents pour leur soutien sans faille tout au long de cette expérience, et leur dis merci.

Et enfin, je tiens à exprimer toute ma gratitude à Jenni Meredith, ma correspondante et amie britannique qui, avec gentillesse et patience, a relu l'intégralité du manuscrit et m'a donné les indications nécessaires à l'amélioration de la précision sémantique. Qu'elle en soit vivement remerciée.

Jean-Charles Bricola

Bibliography

[Angulo López, 2003] Angulo López, J. (2003). *Morphologie mathématique et indexation d'images couleur: application à la microscopie en biomédecine*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris.

[Arbelaez et al., 2011] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916.

[Ayache and Hansen, 1988] Ayache, N. and Hansen, C. (1988). Rectification of images for binocular and trinocular stereovision. In *9th International Conference on Pattern Recognition, 1988.*, pages 11–16. IEEE.

[Aydin and Akgul, 2010] Aydin, T. and Akgul, Y. S. (2010). Stereo depth estimation using synchronous optimization with segment based regularization. *Pattern Recognition Letters*, 31(15):2389–2396.

[Beucher, 1990] Beucher, S. (1990). *Segmentation d'Images et Morphologie Mathématique*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris.

[Beucher, 1998] Beucher, S. (1998). Sets, partitions and functions interpolations. *Journal of Computational Imaging and Vision*, 12:307–314.

[Beucher, 2005] Beucher, S. (2005). Numerical residues. In *Image and Vision Computing*, volume 25, pages 405–415. International Symposium on Mathematical Morphology.

[Beucher, 2011] Beucher, S. (2011). Sur un problème de définition l'érosion géodesique. Technical report, Centre of Mathematical Morphology - Mines ParisTech.

[Beucher, 2012] Beucher, S. (2012). Algorithmic description of erosions and dilations in mamba. Technical report, Centre of Mathematical Morphology - Mines ParisTech.

[Beucher, 2013a] Beucher, S. (2013a). Basic morphological operators applied on partitions. Technical report, Centre of Mathematical Morphology - Mines ParisTech.

[Beucher, 2013b] Beucher, S. (2013b). Maxima and minima: a review. Technical report, Centre of Mathematical Morphology - Mines ParisTech.

[Beucher and Beucher, 2011] Beucher, S. and Beucher, N. (2011). Hierarchical queues: general description and implementation in mamba image library. Technical report, Centre of Mathematical Morphology - Mines ParisTech.

[Beucher and Meyer, 1992] Beucher, S. and Meyer, F. (1992). The morphological approach to segmentation: the watershed transformation. *Optical Engineering*, 34:433–481.

[Bleyer and Gelautz, 2005] Bleyer, M. and Gelautz, M. (2005). A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3):128–150.

[Bleyer et al., 2011] Bleyer, M., Rother, C., Kohli, P., Scharstein, D., and Sinha, S. (2011). Object stereo – joint stereo matching and object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3081–3088. IEEE.

[Bobick and Intille, 1999] Bobick, A. F. and Intille, S. S. (1999). Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200.

[Boykov et al., 2001] 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.

[Bricola et al., 2015] Bricola, J.-C., Bilodeau, M., and Beucher, S. (2015). A multi-scale and morphological gradient preserving contrast. In *14th International Congress for Stereology and Image Analysis*.

[Cigla and Alatan, 2013] Cigla, C. and Alatan, A. A. (2013). Information permeability for stereo matching. *Signal Processing: Image Communication*, 28(9):1072–1088.

[De-Maeztu et al., 2012] De-Maeztu, L., Villanueva, A., and Cabeza, R. (2012). Near real-time stereo matching using geodesic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):410–416.

[Decenciere et al., 1998] Decenciere, E., de Fouquet, C., and Meyer, F. (1998). Applications of kriging to image sequence coding. *Signal processing: image communication*, 13(3):227–249.

[Delhomme, 1976] Delhomme, J.-P. (1976). *Applications de la théorie des variables régionalisées dans les sciences de l'eau*. PhD thesis, Université Pierre et Marie Curie.

[Demarty and Beucher, 1998] Demarty, C.-H. and Beucher, S. (1998). Color segmentation algorithm using an hls transformation. *Computational Imaging and Vision*, 12:231–238.

[Facciolo et al., 2015] Facciolo, G., de Franchis, C., and Meinhardt, E. (2015). Mgm: A significantly more global matching for stereovision. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 90.1–90.12. BMVA Press.

[Fleet and Weiss, 2006] Fleet, D. and Weiss, Y. (2006). Optical flow estimation. In *Handbook of mathematical models in computer vision*, pages 237–257. Springer.

[Fua, 1993] Fua, P. (1993). A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine vision and applications*, 6(1):35–49.

[Gomila, 2001] Gomila, C. (2001). *Mise en correspondance de partitions en vue du suivi d'objets*. PhD thesis, École Nationale Supérieure des Mines de Paris.

[Goshtasby, 2012] Goshtasby, A. (2012). Similarity and dissimilarity measures. In *Image Registration*, Advances in Computer Vision and Pattern Recognition, pages 7–66. Springer London.

[Hanbury and Marcotegui, 2006] Hanbury, A. and Marcotegui, B. (2006). Waterfall segmentation of complex scenes. In Narayanan, P., Nayar, S., and Shum, H.-Y., editors, *Computer Vision – ACCV 2006*, volume 3851 of *Lecture Notes in Computer Science*, pages 888–897. Springer Berlin Heidelberg.

[Hartley and Zisserman, 2004] Hartley, R. and Zisserman, A. (2004). *Multiple view geometry in computer vision*. Cambridge University Press.

[Heo et al., 2008] Heo, Y. S., Lee, K. M., and Lee, S. U. (2008). Illumination and camera invariant stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR.*, pages 1–8. IEEE.

[Hirschmüller, 2008] Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Horaud and Monga, 1995] Horaud, R. and Monga, O. (1995). *Vision par ordinateur: outils fondamentaux*, chapter Vision stéréoscopique. Hermes, 2nd edition.

[Hosni et al., 2010] Hosni, A., Bleyer, M., and Gelautz, M. (2010). Near real-time stereo with adaptive support weight approaches. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 1–8.

[Huq et al., 2013] Huq, S., Koschan, A., and Abidi, M. (2013). Occlusion filling in stereo: Theory and experiments. *Computer Vision and Image Understanding*, 117(6):688–704.

[Jaccard, 1901] Jaccard, P. (1901). Bulletin de la société vaudoise des sciences naturelles. Technical report.

[Joulin et al., 2010] Joulin, A., Bach, F., and Ponce, J. (2010). Discriminative clustering for image co-segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1943–1950. IEEE.

[Kang et al., 2001] Kang, S. B., Szeliski, R., and Chai, J. (2001). Handling occlusions in dense multi-view stereo. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.*, volume 1, pages I–103. IEEE.

[Kiran and Serra, 2014] Kiran, B. R. and Serra, J. (2014). Global–local optimizations by hierarchical cuts and climbing energies. *Pattern Recognition*, 47(1):12–24.

[Kleinberg and Tardos, 2006] Kleinberg, J. and Tardos, É. (2006). *Algorithm design*, chapter Network flows. Pearson Education.

[Lerallut, 2006] Lerallut, R. (2006). *Modélisation et interprétation d'images à l'aide de graphes*. PhD thesis, École Nationale Supérieure des Mines de Paris.

[Linchtenstern, 2013] Linchtenstern, A. (2013). *Kriging methods in spatial statistics*. Bachelor thesis, Technische Universität München, Department of Mathematics, Germany. Pages 52–53.

[Loop and Zhang, 1999] Loop, C. and Zhang, Z. (1999). Computing rectifying homographies for stereo vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.*, volume 1. IEEE.

[McLaren, 1976] McLaren, K. (1976). The development of the cie 1976 (l*a*b*) uniform colour space and colour-difference formula. *Journal of the Society of Dyers and Colourists*, 92(9):338–341.

[Meyer, 1979] Meyer, F. (1979). *Cytologie quantitative et morphologie mathématique*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris.

[Meyer, 2001] Meyer, F. (2001). Hierarchies of partitions and morphological segmentation. In *Scale-Space and Morphology in Computer Vision*, pages 161–182. Springer.

[Meyer, 2004] Meyer, F. (2004). Levelings, image simplification filters for segmentation. *Journal of Mathematical Imaging and Vision*, 20(1-2):59–72.

[Müller, 2007] Müller, M. (2007). Dynamic time warping. In *Information Retrieval for Music and Motion*, pages 69–84. Springer Berlin Heidelberg.

[Ohta and Kanade, 1985] Ohta, Y. and Kanade, T. (1985). Stereo by intra-and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):139–154.

[Prince, 2012] Prince, S. (2012). *Computer vision: models, learning, and inference*, chapter Models for grids, pages 228–247. Cambridge University Press.

[Risson, 2001] Risson, V. (2001). *Application de la morphologie mathématique à l'analyse des conditions d'éclairage des images couleur*. PhD thesis, École Nationale Supérieure des Mines de Paris.

[Rivest et al., 1993] Rivest, J.-F., Soille, P., and Beucher, S. (1993). Morphological gradients. *Journal of Electronic Imaging*, 2(4):326–336.

[Rubio et al., 2012] Rubio, J. C., Serrat, J., López, A., and Paragios, N. (2012). Unsupervised co-segmentation through region matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 749–756. IEEE.

[Scharstein, 1994] Scharstein, D. (1994). Matching images by comparing their gradient fields. In *Pattern Recognition, 1994. Vol. 1. Proceedings of the 12th IAPR International Conference on Computer Vision and Image Processing*, volume 1, pages 572–575. IEEE.

[Scharstein et al., 2014] Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nesic, N., Wang, X., and Westling, P. (2014). High-resolution stereo datasets with subpixel-accurate ground truth. In Jiang, X., Hornegger, J., and Koch, R., editors, *GCPR*, volume 8753 of *Lecture Notes in Computer Science*, pages 31–42. Springer.

[Scharstein and Szeliski, 1998] Scharstein, D. and Szeliski, R. (1998). Stereo matching with nonlinear diffusion. *International journal of computer vision*, 28(2):155–174.

[Scharstein and Szeliski, 2002] Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*.

[Serra, 1983] Serra, J. (1983). *Image analysis and mathematical morphology*. Academic Press, Inc.

[Sinha et al., 2014] Sinha, S. N., Scharstein, D., and Szeliski, R. (2014). Efficient high-resolution stereo matching using local plane sweeps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1582–1589. IEEE.

[Sun et al., 2005] Sun, J., Li, Y., Kang, S. B., and Shum, H.-Y. (2005). Symmetric stereo matching for occlusion handling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, volume 2, pages 399–406. IEEE.

[Szeliski, 2011] Szeliski, R. (2011). *Computer Vision: Algorithms and Applications*, chapter Image processing, section 3.2.1 on separable filtering. Texts in Computer Science. Springer London.

[Twardowski et al., 2004] Twardowski, T., Cyganek, B., and Borgosz, J. (2004). Gradient based dense stereo matching. In *Image Analysis and Recognition*, pages 721–728. Springer.

[Vachier and Meyer, 2005] Vachier, C. and Meyer, F. (2005). The viscous watershed transform. *Journal of Mathematical Imaging and Vision*, 22(2-3):251–267.

[Vachier and Vincent, 1995] Vachier, C. and Vincent, L. (1995). Valuation of image extrema using alternating filters by reconstruction. In *SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation*, pages 94–103.

[Vilaplana et al., 2008] Vilaplana, V., Marques, F., and Salembier, P. (2008). Binary partition trees for object detection. *IEEE Transactions on Image Processing*, 17(11):2201–2216.

[Vincent and Dougherty, 1994] Vincent, L. and Dougherty, E. R. (1994). Morphological segmentation for textures and particles. *Digital image processing methods*, 42:43–102.

[Yamaguchi et al., 2012] Yamaguchi, K., Hazan, T., McAllester, D., and Urtasun, R. (2012). Continuous markov random fields for robust stereo estimation. In *European Conference on Computer Vision 2012 Proceedings*, pages 45–58. Springer.

[Yang et al., 2009] Yang, Q., Wang, L., Yang, R., Stewénius, H., and Nistér, D. (2009). Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3).

[Zabih and Woodfill, 1994] Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. In *Third European Conference on Computer Vision 1994 Proceedings, Volume II*, pages 151–158. Springer Berlin Heidelberg.

[Zanoguera Tous, 2001] Zanoguera Tous, M. F. (2001). *Segmentation interactive d'images fixes et de séquences vidéo basée sur des hiérarchies de partitions*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris.

[Zbontar and LeCun, 2015] Zbontar, J. and LeCun, Y. (2015). Stereo matching by training a convolutional neural network to compare image patches. *arXiv preprint 1510.05970*.

[Zitnick and Kang, 2007] Zitnick, C. L. and Kang, S. B. (2007). Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75(1):49–65.

Résumé

Cette thèse propose de nouvelles approches pour le calcul de cartes de profondeur associées à deux images stéréoscopiques.
La difficulté du problème réside dans l'établissement de mises en correspondances entre les deux images stéréoscopiques. Cet établissement s'avère en effet incertain dans les zones de l'image qui sont homogènes, voire impossible en cas d'occultation.
Afin de gérer ces deux problèmes, nos méthodes procèdent en deux étapes. Tout d'abord nous cherchons des mesures de profondeur fiables en comparant les deux images stéréoscopiques à l'aide de leurs segmentations associées. L'analyse des coûts de superpositions d'images, sur une base régionale et au travers d'échelles multiples, nous permet de réaliser des agrégations de coûts pertinentes, desquelles nous déduisons des mesures de disparités précises. De plus, cette analyse facilite la détection des zones de l'image de référence étant potentiellement occultées dans l'autre image de la paire stéréoscopique. Dans un deuxième temps, un mécanisme d'estimation se charge de trouver les profondeurs les plus plausibles, là où aucune mise en correspondance n'a pu être établie.
L'ouvrage est scindé en deux parties : la première permettra au lecteur de se familiariser avec les problèmes fréquemment observés en analyse d'images stéréoscopiques. Il y trouvera également une brève introduction au traitement d'images morphologique. Dans une deuxième partie, nos opérateurs de calcul de profondeur sont présentés, détaillés et évalués.

Mots Clés

Traitement d'image morphologique, cartes de disparités, vision stéréoscopique, analyse d'images stéréo basée sur la segmentation, gestion des occultations, agrégation de coûts, filtrage de bavures, interpolation, approches multi-échelles.

Abstract

In this thesis, we introduce new approaches dedicated to the computation of depth maps associated with a pair of stereo images.
The main difficulty of this problem resides in the establishment of correspondences between the two stereoscopic images. Indeed, it is difficult to ascertain the relevance of matches occurring in homogeneous areas, whilst matches are infeasible for pixels occluded in one of the stereo views.

In order to handle these two problems, our methods are composed of two steps. First, we search for reliable depth measures, by comparing the two images of the stereo pair with the help of their associated segmentations. The analysis of image superimposition costs, on a regional basis and across multiple scales, allows us to perform relevant cost aggregations, from which we deduce accurate disparity measures. Furthermore, this analysis facilitates the detection of the reference image areas, which are potentially occluded in the other image of the stereo pair. Second, an interpolation mechanism is devoted to the estimation of depth values, where no correspondence could have been established.

The manuscript is divided into two parts: the first will allow the reader to become familiar with the problems and issues frequently encountered when analysing stereo images. A brief introduction to morphological image processing is also provided. In the second part, our algorithms to the computation of depth maps are introduced, detailed and evaluated.

Keywords

Morphological image processing, disparity maps, stereovision, segmentation-based stereo image analysis, occlusion handling, cost aggregation, filtering of fattening artefacts, interpolation, multi-scale approaches.