HAL
open science

Connected image processing with multivariate attributes: an unsupervised Markovian classification approach

Benjamin Perret, Christophe Collet

▶ To cite this version:

Benjamin Perret, Christophe Collet. Connected image processing with multivariate attributes: an unsupervised Markovian classification approach. Computer Vision and Image Understanding, 2015, 133, pp.1-14. 10.1016/j.cviu.2014.09.008 . hal-01071418

HAL Id: hal-01071418
https://hal.science/hal-01071418v1

Submitted on 10 Oct 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Connected image processing with multivariate attributes: an unsupervised Markovian classification approach

B. Perret[a,*], Ch. Collet[b]

[a] *Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge UMR 8049, ESIEE Paris, CNRS, France*
[b] *iCube lab., Université de Strasbourg - UMR CNRS 7357, France*

Abstract

This article presents a new approach for constructing connected operators for image processing and analysis. It relies on a hierarchical Markovian unsupervised algorithm in order to classify the nodes of the traditional Max-Tree. This approach enables to naturally handle multivariate attributes in a robust non-local way. The technique is demonstrated on several image analysis tasks: filtering, segmentation, and source detection, on astronomical and biomedical images. The obtained results show that the method is competitive despite its general formulation. This article provides also a new insight in the field of hierarchical Markovian image processing showing that morphological trees can advantageously replace traditional quadtrees.

Keywords: Image filtering, image segmentation, connected filter, Max-Tree, connected component tree, hierarchical modeling, Markov model, mathematical morphology

1. Introduction

In image processing, connected operators are morphological operators that concentrate on the non deformation of edges. In binary images, as their name suggests, their very principle is to work on the connected components of an image, and the only allowed operation is the removal of such components. Connected operators take their roots in the notion of filters by reconstruction [1, 2]. They have been studied in the context of binary image processing in [3, 4] and their extension to gray-scale images appeared in [5, 6, 7, 8]. In this case, we say that an operator is connected if it is connected at all thresholding levels of the image.

Gray-scale connected operators started to become popular when a powerful framework and an efficient algorithm were proposed in order to compute connected operators by Salembier *et al.* [9]. This framework relies on a hierarchical representation of the image called the *Max-Tree* (also commonly known as the *(connected) component tree*). This representation is a tree where each node corresponds to a connected component of a threshold of the image and the parent relation is given by the inclusion relation among the connected components. This representation can then be used in the following 4-steps process (see Fig. 1) to define connected operators: 1) compute the Max-Tree of the image, 2) equip each node of the tree with some relevant attributes (area, compactness, moments, entropy...), 3) select nodes in the tree according to their attribute values, and finally 4) reconstruct an image from the filtered tree. The last reconstruction step consists in assigning a new value to each pixel of the image using the content of the nodes selected during the previous step. Connected operators have since been applied to various types of images: document images [10, 11, 12], biomedical imaging [13, 14, 15, 16], remote sensing [17, 18], or astronomical imaging [19, 20]. Although the primary aim of connected operators was to perform image filtering [9], they have now been implied in various image analysis tasks such as segmentation [8, 21, 22], retrieval [23], classification [24] or registration [25].

It is also noteworthy that several connected operators (morphological reconstruction, flooding, region growing...) can be formulated in the framework of the Image Forestring Transform (IFT) [26]. The IFT relies on an efficient and versatile formulation based on the classical shortest path problem and provides a unified framework for a wide variety of image operators. On the other hand it has also been observed that connected operators share deep links with the TV-L1 optimization scheme [27].

1.1. Related works

Researches have been carried out in order to improve the capacity or the efficiency at each step of the method.

About the construction of the representation, a first category of works concentrates on the definition of the connected components: connections [28], second-generation

*Corresponding author. Tel: +331 45 92 67 09, Address: ESIEE Paris, 2 boulevard Blaise Pascal, Cité Descartes, BP 99, 93162 Noisy le Grand Cedex France

Email addresses: benjamin.perret@esiee.fr (B. Perret), c.collet@unistra.fr (Ch. Collet)

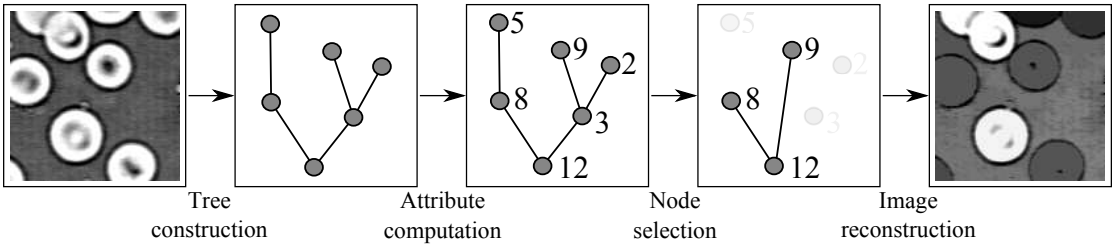


Figure 1: The general 4 step process used to construct a connected operator: 1) construct the Max-Tree representing the image, 2) compute node attributes, *i.e.*, attach features to each node, 3) select nodes based on their attribute value: the selected nodes form a new tree, and 4) reconstruct an image from the new tree, *i.e.*, assign a value to each pixel based on the remaining nodes in the filtered tree.

connections [29, 30, 14, 31], hyperconnections [12], or directed connections [32]. The relations among various definitions of connections have been studied in [33].

Other works proposed to replace the Max-Tree by other representations, generally designated by *morphological trees*: binary partition trees [34], hierarchies induced by constrained connectivities [35], hierarchies of minimum spanning forests [36], the tree of shapes [37, 38], the connectivity tree [39], the alpha-tree [40]... One can note that the relations among some of these representations have been studied in [41, 42].

The use and definition of relevant node attributes have also been studied in various works. Among the proposed attributes, we find: area [43, 44], dynamic [45], simplicity and entropy [9], various geometrical moments and tensors [13, 21], multispectral volume and color [20], contrast [10], Mumford-Shah energy [46] and so on.

While most connected operators designed so far rely on a simple thresholding on the attribute values – *i.e.*, a node is selected if its attribute value is larger than a specified threshold – it has immediately been noted that such a purely local strategy tends to to be sensitive to noise whenever the attributes is not increasing: *i.e.*, when the attribute value does not vary monotonically along a branch of the tree. In the seminal work of Salembier *et al.* [9] the authors proposed several regularization strategies that all rely on the idea that the selection process should realize a pruning of the tree. The three proposed strategies were the followings:

- max: a branch is pruned at its highest selected node;
- min: a branch is pruned at its lowest selected node;
- Viterbi: an energy criterion is optimized in order to determine the best pruning level.

Recently, Xu *et al.* proposed in [16, 47] to apply another connected filter on the tree representation seen as an image whose adjacency relation is given by the parent-children relation in the tree, and whose pixel values are the attribute values. This approach provides a non local strategy to detect relevant changes in the attribute values.

Another difficulty of the node selection process is the handling of multiple or multivariate attributes. In [48] Urbach *et al.* proposed two strategies to handle multivariate attributes: the direct extension of the thresholding approach where a node is selected if each of its attributes is greater than a particular threshold (equivalent to a marginal ordering of the attribute space), or using a distance to a reference attribute vector where a node is selected if the distance from its attribute vector to a reference vector is lower than a given threshold (equivalent to a reduced ordering of the attribute space). In both cases, there is a need to define a vector ordering which is a non trivial problem. In [49], the authors proposed to learn the distance function used to compare vector attributes from the distribution of the attributes of the nodes of interest. They modeled this distribution as a multivariate Gaussian, then learn the parameters of the distribution using a ground truth, and finally select the nodes with a threshold on the Mahalanobis distance defined by the model parameters. Although this approach was a first attempt to introduce some learning on multivariate data in the construction of connected operators, the node selection process still remained purely local.

Finally, the reconstruction step has been initially studied in [9] which defines a *natural* reconstruction: a pixel is valued with the level of the smallest selected node that contains it. Urbach *et al.* proposed in [24] to alleviate the problem of non increasing attributes by modifying the reconstruction process with the *subtractive* rule. In this strategy, whenever a node is removed, the level of its descendants are lowered by the height of the deleted node. This strategy has been used for the production of shape-size pattern spectra. Finally, in [11] the authors proposed a reconstruction process based on the notion of hyperconnection that enables the preservation of texture details during the reconstruction process.

1.2. Contributions

In this article, we propose a novel approach in order to select nodes in the Max-Tree. Our idea is to perform an unsupervised Bayesian classification of the nodes using a hierarchical Markovian model. This approach provides several benefits:

- a natural handling of multiple or multivariate attributes through the use of multivariate probability density functions;

- a robust global process for the classification of the nodes;
- an unsupervised approach that learns the model from each image without any ground truth.

Hierarchical Markovian models exist for about two decades now [50]. The advantage of these models compared to traditional hidden Markov random fields [51, 52] is the possibly of computing exact inference without iterative algorithms. They have since been extensively applied to quadtrees [53, 54] or similar structures [55]. In the same way, Hidden Markov Chains (HMC) allow an exact inference based on Maximum of Posterior Marginal (MPM) criterion and both approaches on chains [56, 57] and quadtrees [58] have been refined through pairwise or triplet models in the past decade. Indeed, Markovian chains share similar properties with Markovian quadtrees than can be seen as hierarchical Markov chains [59].

In the proposed scheme, we use the Max-Tree to define a hierarchical Markovian probabilistic model. The attribute values of the nodes serve as observations, and we aim at finding the most probable hidden label value of each node of the tree. The existing iterative unsupervised classification algorithms for quadtree-like structures remain adapted to this model.

We propose several experiments in order to demonstrate how this approach can be used in different image analysis tasks: shape separation in artificial images, source detections in astronomical images, and vessel network segmentation in eye fundus images are some examples given in this paper to show the generic ability of this model. In those experiments, we study the use and combination of various node attributes and how to exploit the classification results in order to obtain a filtering, a segmentation, or extract sources. The results obtained on source detection are compared to those obtained with existing methods and the vessel network segmentation method is evaluated on two standard image databases.

The contribution of this article can also be considered from the field of hierarchical Markovian image processing where the replacement of the usual quadtree model by a morphological tree provides several benefits:

- morphological trees are naturally adapted to each image, removing the presence of the block artifacts of the quadtree [54];
- they are more compact than quadtrees (less nodes in the tree) and can easily be preprocessed in order to drastically decrease the number of nodes which leads to better computation time;
- they offer the possibility to integrate various node attributes as multivariate observations. More specifically, all the shape descriptors that have no meaning in a quadtree are now usable; and
- observations are naturally present at every level of the tree, whereas we usually only have data at the leave level in a quadtree in practice.

This manuscript is organized as follows. Section 2 presents the Max-Tree, gives the definition of several node attributes, and explains how to reconstruct an image from the tree. Section 3 describes the probabilistic model used to classify the nodes of the tree, and the unsupervised classification algorithm (given in AppendixA). The necessary pre-processing of the data is explained in Section 4 whereas some experiments and results are detailed in Section 5. Finally, Section 6 concludes this work.

2. Max-Tree

We first give some basic notations and definitions that will be used throughout the article.

We represent the *image domain* by a non empty finite subset E of $\mathbb{Z}^2$. An element of E is called *a pixel, or a point*. A *gray-scale image* f is a function from the domain E to the set of real numbers $\mathbb{R}$. The value of the gray-scale image f at pixel x, denoted by $f(x)$, is called the *level* of x in f.

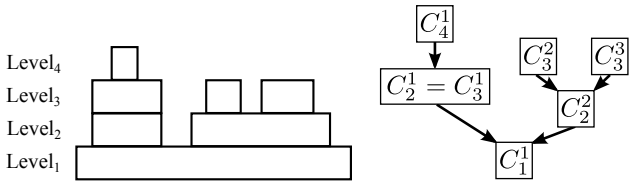
2.1. Definition of the tree

The Max-Tree of an image is based on the decomposition of every possible threshold of the image into connected components. Connected components may be defined by any appropriate mean: *e.g.*, by path connectivity in a graph or a general connection on the set of all subsets of E. In this article, all the examples are based on a classical 8 neighborhood on a regular square grid.

Let $X \subseteq E$ be a set of points, we denote by $\mathrm{CC}_x(X)$ the connected component of X that contains the pixel x. We denote by $\mathrm{CC}(X)$ the set of connected components of X: $\mathrm{CC}(X) = \{\mathrm{CC}_x(X),\ x \in X\}$. Moreover, we assume that E is connected: *i.e.*, for any x in E, we have $\mathrm{CC}_x(E) = E$.

Given an image f, the *level set* of f at level k in $\mathbb{R}$, denoted by $\mathrm{Level}_k(f)$, is the set of pixels where the value of f is larger than or equal to k: *i.e.*, $\mathrm{Level}_k(f) = \{x \in E \,|\, f(x) \geq k\}$. We denoted by $\mathrm{CC}(f)$ the set of all connected components of all level sets of f: *i.e.*, $\mathrm{CC}(f)$ is equal to $\bigcup_{k\in\mathbb{R}} \mathrm{CC}(\mathrm{Level}_k(f))$. It is well known that any two elements of $\mathrm{CC}(f)$ are either disjoint or comparable: *i.e.*, for any A and B in $\mathrm{CC}(f)$, $A \cap B \neq \emptyset$ implies either $A \subseteq B$ or $B \subseteq A$.

Then, we can define the parent mapping on $\mathrm{CC}(f)$, denoted by $\mathrm{par}_{\mathrm{CC}(f)}$, that associates to any element n of $\mathrm{CC}(f)$ its *parent* $\mathrm{par}_{\mathrm{CC}(f)}(n)$, also denoted by n^-, and defined as the smallest element of $\mathrm{CC}(f)$ that is strictly larger than n if it exists or *undefined* otherwise. For the sake of concision, we will simply write par instead of $\mathrm{par}_{\mathrm{CC}(f)}$ when no confusion is possible. The *Max-Tree* of f is then the tree defined by the set of nodes $\mathrm{CC}(f)$ and the parent mapping $\mathrm{par}_{\mathrm{CC}(f)}$. In this context, the elements of $\mathrm{CC}(f)$ are called *nodes*. The single node n in $\mathrm{CC}(f)$ such



(a) 1d image f (b) Max-Tree of f

Figure 2: (a) represents a 1d image f with 4 levels. (b) is the Max-Tree of the image f. C_i^j denotes the j-th connected component of the i-th level set of f (components in a given level are arbitrarily numbered from left to right). The arrows represent the parent relation on the elements of $\mathrm{CC}(f)$. One can note that the two components C_2^1 and C_3^1 are equal and are thus represented by a single node in the tree.

that n^- is equal to *undefined* is the *root* of the tree. This construction is illustrated in Fig. 2.

Several efficient data-structures and algorithms ($\mathcal{O}(n)$ or $\mathcal{O}(n\log(n))$ time complexity depending on the type of data processed) have been proposed to compute the Max-Tree of an image: see [9, 60] and [61] for a survey of the different algorithms.

Let n and m be two nodes in $\mathrm{CC}(f)$. If the node n is the parent of the node m then we say that m is a child of n, and we denote by n^+, the set of all children of n. If there exists a positive integer k such that $n = \mathrm{par}^k(m) = \underbrace{\mathrm{par}(\ldots(\mathrm{par}}_{k \text{ times}}(m)\ldots))$, *i.e.*, if n is the parent of the parent of... of m, we say that n is an ancestor of m and that m is a descendant of n. Given a node n, the set of descendants of n is denoted $n>$.

One can note that any non empty subset of nodes of any Max-Tree still verifies the *disjoint or comparable* property and defines thus a sub-tree of (or a forest). In the following, given a subset of A of $\mathrm{CC}(f)$, we define *the tree induced by* A by the tree whose set of nodes if A and whose parent relation is par_A. Indeed, one can show that the tree induced by A is the Max-Tree of a filtered version of f: this tree can be processed as any other Max-Tree.

2.2. Node attributes

The nodes of the Max-Tree of an image are usually equipped with attributes, *i.e.*, features that are relevant for a given application. For example, if one is looking for vessels in a medical image, a feature measuring the elongation of the node may be a good point to start with.

We define hereafter some node attributes that will be used in the following of this article. Let f be a gray-scale image, and let $n \subseteq E$ be a node in $\mathrm{CC}(f)$.

$\mathrm{Area}(n)$ is the number of pixels in n: $\mathrm{Area}(n) = \mathrm{card}(n)$;

$\mathrm{Level}(n)$ the level of n is the largest level k in $\mathbb{R}$ such that n belongs to the level set of f at level k: $\mathrm{Level}(n) = \max\{\ell \in \mathbb{R} \mid n \in \mathrm{Level}_\ell(f)\}$. By extension, we set $\mathrm{Level}(undefined)$ equals $\min\{f(x),\ x \in E\}$.

$\mathrm{Highest}(n)$ is the level of the highest node in the branch rooted in n:

$$\mathrm{Highest}(n) = \max\{\mathrm{Level}(m), m \in \mathrm{CC}(f) \mid n \text{ is an ancestor of } m\};$$

$\mathrm{Depth}(n)$ is the difference between the level of the highest node in the branch rooted in n and the level of the parent of n: $\mathrm{Depth}(n) = \mathrm{Highest}(n) - \mathrm{Level}(n^-)$;

$\mathrm{Perimeter}(n)$ is the number of pixels of $E \backslash n$ that are adjacent to n;

$\mathrm{Circularity}(n)$ is the ratio of the area of n and the square of its perimeter:

$$\mathrm{Circularity}(n) = \mathrm{Area}(n)/(\mathrm{Perimeter}(n))^2.$$

The circularity measure is translation, rotation, and scale invariant (up to the quantization effects). It is equal to $1/(4\pi)$ for a circle which is the most compact shape, and it decreases as the shape goes off the perfect circle;

$\bar{x}(n)$ **and** $\bar{y}(n)$ are the x and y coordinate of the centroid of the shape n: $\bar{x}(n)$ is equal to $(\sum_{(x,y)\in n} x)/\,\mathrm{card}(n)$ and $\bar{y}(n)$ is equal to $(\sum_{(x,y)\in n} y)/\,\mathrm{card}(n)$;

$\mu_{pq}(n)$ the central moment of order pq: $\mu_{pq}(n)$ is equal to $\sum_{(x,y)\in n}(x-\bar{x}(n))^p(y-\bar{y}(n))^q$;

$\mathrm{Elongation}(n)$ the elongation of n obtained from the central moments: $\mathrm{Elongation}(n)$ is equal to $\sqrt{\lambda_2/\lambda_1}$ with

$$\lambda_i = \mu_{20}(n) + \mu_{02}(n) \pm \sqrt{4\mu_{11}^2(n) + (\mu_{20}(n) - \mu_{02}(n))^2};$$

$\mathrm{Hu}_1(n)$: the first moment of Hu [62] of the shape n equivalent to a moment of inertia, it is translation, rotation, and scale invariant: $\mathrm{Hu}_1(n)$ is equal to $(\mu_{20}(n) + \mu_{02}(n))/\,\mathrm{card}^2(n)$.

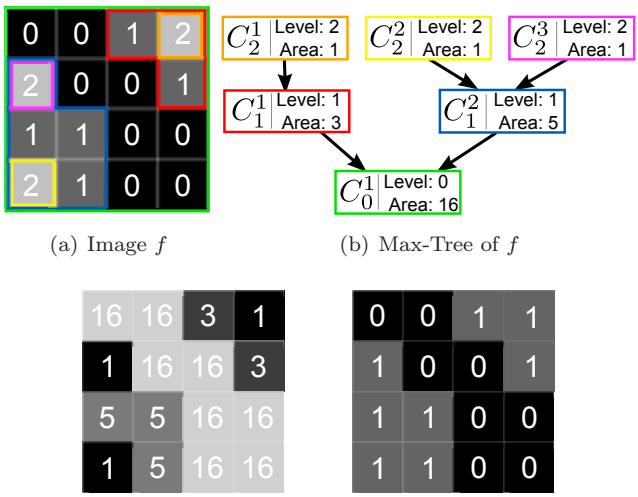
All these attributes can be computed efficiently either during the construction of the tree or as a linear post-processing.

2.3. Filtering and tree reconstruction

In order to filter an image using the Max-Tree, one must be able to map a tree back into the image space: this step is called a reconstruction. The reconstruction procedure involves: 1) to define the nodes of the tree that will be reconstructed: this is given by a selection criterion, and 2) to define which value will be assigned to the reconstructed nodes: this is a valuation function.

Formally, a *criterion* σ on $\mathrm{CC}(f)$ is a function that associates the value true or false to each node in $\mathrm{CC}(f)$. Given a node n, we say that n *satisfies* σ if $\sigma(n) = \text{true}$.

For example, given a size threshold $t \in \mathbb{R}^+$, a simple criterion is σ_a defined for any node $n \in \mathrm{CC}(f)$ by $\sigma_a(n) = \text{true}$ if $\mathrm{Area}(n) > t$ and false otherwise. This criterion is satisfied by the nodes whose area is larger than t. By abuse of notation, we also denote by true (resp. false), the criterion that associates the value true (resp. false) to every node in $\mathrm{CC}(f)$.



(a) Image f (b) Max-Tree of f

(c) Attribute Area (d) Area filter

Figure 3: (a) represents a 4×4 image f with 3 levels. The contours of the connected components of the different levels of f are depicted with different colors. (b) is the Max-Tree of the image f. Each component C_i^j (the j-th connected component of the i-th level of f) is depicted with the same color as the one used to draw its contour in (a). The values of the two attributes Area and Level are also indicated in each component. (c) represents the reconstruction of the attribute Area ($\mathrm{R}[\mathrm{CC}(f), \text{true}, \text{Area}]$): the value of each pixel has been replaced by the area of the smallest component containing this pixel. (d) represents the reconstruction of the attribute Level of the nodes larger than 2 pixels ($\mathrm{R}[\mathrm{CC}(f), \text{Area} > 2, \text{Level}]$): the value of each pixel has been replaced by the level of the smallest component containing this pixel and larger than 2 pixels.

A *valuation* v on $\mathrm{CC}(f)$ is a function that associates a value to each node in $\mathrm{CC}(f)$: *i.e.*, v maps $\mathrm{CC}(f)$ to $\mathbb{R}^d$ with d a strictly positive integer. The mapping Level that associates its level to any node is a natural valuation, but any attribute value can be used.

Finally, the *reconstruction* of the Max-Tree constrained by the criterion σ for the valuation v, is an image from E to $\mathbb{R}^d$ denoted by $\mathrm{R}[\mathrm{CC}(f), \sigma, v]$. Given a pixel x of E, the value of $\mathrm{R}[\mathrm{CC}(f), \sigma, v]$ at pixel x is the valuation of the smallest node in $\mathrm{CC}(f)$ that contains x and that satisfies σ:

$$\mathrm{R}[\mathrm{CC}(f), \sigma, v](x) = v(\min\{n \in \mathrm{CC}(f) \,|\, x \in n \text{ and } \sigma(n) = \text{true}\}). \quad (1)$$

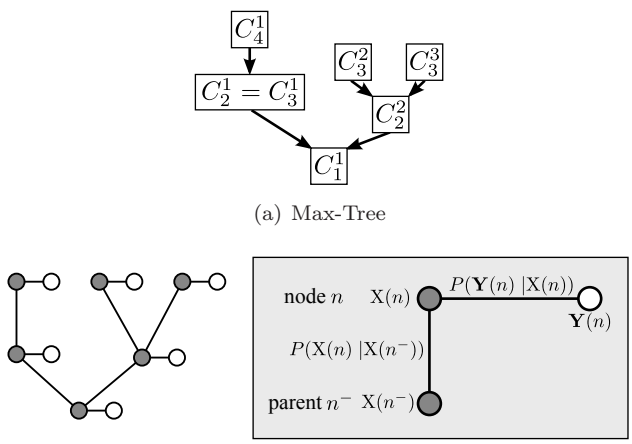
Two examples of reconstructions are given in Fig. 3. It can be shown that the reconstruction of the whole (using the criterion true) Max-Tree of the function f with the valuation Level is equal to f itself: $\mathrm{R}[\mathrm{CC}(f), \text{true}, \text{Level}] = f$. In other words, if we reconstruct the tree without any filtering using the initial level values of the components we get the original image back. Also note that the root node must satisfy σ in order to guaranty that Eq.(1) is always correctly defined.

3. Bayesian tree classification

In the following, given an image f from E to $\mathbb{R}$, we consider the problem of classifying the nodes of the Max-Tree of f into K classes using a Bayesian approach. In this context, we consider that each node owns two values: an observation and a label value. The label value is hidden and we want to determine its more probable value according to a probabilist model that defines how observations are linked to label values (likelihood) and the assumptions we have on the distribution of the labels (prior). Note that in the context of Bayesian classification the nodes of the tree are usually called *sites*.

The labels are represented by the mapping X from the set of nodes $\mathrm{CC}(f)$ to the set of labels $\Omega_X = 1, \ldots, K$. The observations are represented by the mapping $\mathbf{Y}$ from the set of nodes $\mathrm{CC}(f)$ to $\mathbb{R}^d$. Our goal is then, for each node n in $\mathrm{CC}(f)$, to find the label ℓ_n that maximizes the marginal posterior probability of having $\mathrm{X}(n)$ equals ℓ_n knowing $\mathbf{Y}$:

$$\ell_n = \arg\max_{k \in \Omega_X} P(\mathrm{X}(n) = k \,|\, \mathbf{Y}) \quad (2)$$



(a) Max-Tree

(b) Hidden markov model

Figure 4: (a) The Max-Tree of the Figure 2. (b) The associated hierarchical hidden Markov model. The structure of the Markovian model is exactly the one of the Max-Tree. To each node n of the tree corresponds a pair composed of a hidden label $\mathrm{X}(n)$ (gray circle) and an observation $\mathbf{Y}(n)$ (white circle). The probability of observing the data $\mathbf{Y}(n)$ on the node n depends of its label value $\mathrm{X}(n)$. The probability of having the label $\mathrm{X}(n)$ at the node n depends of the label of its parent $\mathrm{X}(n^-)$.

To simplify notation, we denote the discrete probability and the probability density function with the same symbol P. Whenever the situation is not ambiguous we write $P(X)$ for $P(X = x)$.

3.1. Hierarchical Markov model

The idea is to use the Max-Tree of f to define a hierarchical Markov model on the nodes $\mathrm{CC}(f)$. This construction is illustrated in Fig. 4. In such a model:

- the probability of observing the data $\mathbf{y}$ on the node n only depends of the label of n: $P(\mathbf{Y}(n) = y \mid \mathrm{X}) = P(\mathbf{Y}(n) = y \mid \mathrm{X}(n))$;
- the probability of having the label k on the non root node n conditional on all forefathers of n only depends of the label of the father of n: $P(\mathrm{X}(n) = k \mid \mathrm{X}) = P(\mathrm{X}(n) = k \mid \mathrm{X}(n^-))$; and
- for the root node r, we have $P(\mathrm{X}(r) = k \mid \mathrm{X}) = P(\mathrm{X}(r) = k)$.

This model provides a hierarchical Markovian regularization in the classification process as the label of a node only depends of the label of its parent.

Under these hypothesis, the joint probability of X and $\mathbf{Y}$ can be factorized as:

$$P(\mathrm{X}, \mathbf{Y}) = P(\mathrm{X}(r)) \times \prod_{\substack{n \in \mathrm{CC}(f) \\ n \neq r}} P(\mathrm{X}(n) \mid \mathrm{X}(n^-)) \times \prod_{n \in \mathrm{CC}(f)} P(\mathbf{Y}(n) \mid \mathrm{X}(n)) \quad (3)$$

with r the root node.

It is well known that in such an acyclic model, the posterior marginal probabilities (2), and thus the label of each node, can be computed with a deterministic belief message passing algorithm [63]. This algorithm requires two traversals of the tree:

- the top-down pass, where the tree is browsed from the leaves to the root, computes for each node n in $\mathrm{CC}(f)$, the posterior probability of the label $\mathrm{X}(n)$ conditionally to the observations available in the descendants of n: $P(\mathrm{X}(n) \mid \mathbf{Y}(m), m \text{ is a descendant of } n)$;
- the bottom-up pass, where the tree is browsed from the root to the leaves, computes for each node n in $\mathrm{CC}(f)$, the posterior probability of the label $\mathrm{X}(n)$ conditionally to the observations $\mathbf{Y}$: $P(\mathrm{X}(n) \mid \mathbf{Y})$.

The detailed procedure is given in Algorithm 1 in appendix. One can note that in the context of quadtree processing, trees being represented with the root at the top, the "top-down pass" is called "bottom-up pass" (and conversely).

However, this in-scale procedure assumes that the model parameters (root probabilities, transition probabilities, and data-driven likelihoods) are known and a more complex iterative algorithm, presented in the next section, must be used in order to obtain an unsupervised classification strategy when those parameters also have to be estimated.

3.2. Unsupervised classification algorithm

In this section, we present an unsupervised classification algorithm which is a direct adaptation of the algorithms described in [53, 54] and the reader can refer to those articles for a detailed description.

We only consider the case of Gaussian likelihoods for the data-driven term. Thus, for any n in $\mathrm{CC}(f)$ and any class k, we have:

$$P(\mathbf{Y}(n) \mid \mathrm{X}(n) = k) = \mathcal{N}(\mathbf{Y}(n); \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (4)$$

with $\mathcal{N}(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ the multivariate Gaussian distribution of mean $\boldsymbol{\mu}_k$ in $\mathbb{R}^d$ and of $d \times d$ covariance matrix $\boldsymbol{\Sigma}_k$.

Then, the transition probability from a label to another one, *i.e.*, the prior probability of having the label k of Ω_X on a non-root node in $\mathrm{CC}(f)$ knowing the label k' of its father, is the same for all nodes:

$$P(\mathrm{X}(n) = k \mid \mathrm{X}(n^-) = k') = t_{k'k} \quad (5)$$

with $t_{k'k}$ a real number in the interval $[0, 1]$ such that $\sum_{k \in \Omega_X} t_{k'k} = 1$. Usually, we have for any k in Ω_X, that t_{kk} is close to 1 which means that, the prior probability that a node has the same label that its parent is high.

Finally, for the root node r in $\mathrm{CC}(f)$ and any label k, we set

$$P(\mathrm{X}(r) = k) = \pi_k \quad (6)$$

in the interval $[0, 1]$ with the constraint $\sum_{k \in \Omega_X} \pi_k = 1$.

The parameters of the model are thus:

- K mean vectors $\boldsymbol{\mu}_k$ of dimension d;
- K covariance matrices $\boldsymbol{\Sigma}_k$ of dimension $d \times d$;
- K prior probabilities π_k for the label of the root node;
- one $K \times K$ transition probability matrix $(t_{k'k})$.

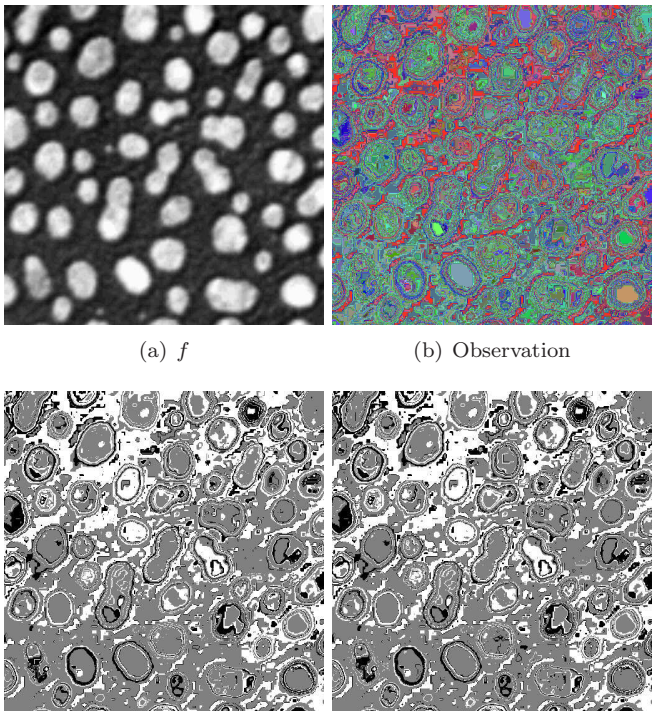
The number of classes K is assumed to be constant and will not be estimated by the algorithm.

The model parameters can be learned from the observations using a two steps iterative procedure:

1. Compute the label estimates with the message passing algorithm according to the current model parameters;
2. Update the model parameters with the Expectation-Maximization (EM) algorithm according to the labels estimated in the previous step.

This leads to a completely unsupervised classification procedure. One can note that this algorithm does not guaranty the optimality of the solution as the EM algorithm may get trapped in a local maximum.

The overall procedure is described in Algorithm 2 in appendix. The initialization of the model parameters with K classes is based on the estimation of a Gaussian mixture



(a) f (b) Observation

(c) Hidden labels (d) Estimated labels

Figure 5: (a) Validation of the segmentation process on the Max-Tree of the image f. Figs (b) and (c) are respectively the reconstruction of the observations and of the hidden labels obtained with a simulation of the random process described in Section 3.1 on the Max-Tree of f. Fig. (d) is the reconstruction of the estimated labels with the unsupervised segmentation algorithm described in Section 3.2 based on the observations (b).

model with K components on the whole set of observations.

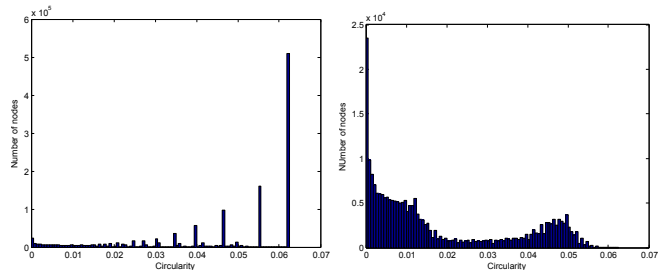
One iteration of the algorithm runs in $\mathcal{O}(N \times K^3)$ time complexity, with N the number of nodes in the tree and K the number of classes. In practice, the algorithm converges in a few iterations (less than 10) which is coherent with the convergence rate usually observed [53, 54, 59]. The Max-Tree has at most as many nodes as the number of pixels in the image, and experiments suggest that, for natural images, the number of nodes in the Max-Tree is of the same order than the number of pixels.

In order to validate our implementation, we have simulated random labels and observations on a Max-Tree of an arbitrary image f according to the hierarchical Markov process described in Section 3.1. The image f (Fig. 5(a)) is a traditional image of blood. Its Max-Tree has 17883 nodes. We have simulated a random process with 3 classes and observations in $\mathbb{R}^3$. Fig. 5(b) and 5(c) are respectively the reconstructed images of the sampled observations $(\mathrm{R}[\mathrm{CC}(f), \mathrm{true}, \mathbf{Y}])$ and labels $(\mathrm{R}[\mathrm{CC}(f), \mathrm{true}, \mathrm{X}])$. Finally, Fig. 5(d) is the reconstruction of the estimated labels given by the unsupervised procedure described in Section 3.2. We see that the estimated labels are close to the original labels, there are indeed 799 misclassified nodes (error rate of 4.44%).

The algorithm has been implemented in Java and was executed on a 2 GHz Core I7 processor. On the previous simulated example, the algorithm converges in 2 iterations with a total running time of 500 ms. The rapid convergence here is not surprising as the data have been generated with the direct model and thus they exactly fit the hypothesis of the algorithm. On the more complex scenario of astronomical image processing (Section 5.2), the 180 000 nodes of the tree are classified in 9 classes: in this case, 1 iteration of the algorithm takes about 6 seconds and the algorithm converges in 3 iterations.

4. Data Preparation

Processing the tree space is just like processing any signal and some pre-processing may be required. Nevertheless, in this case, pre-processing may be performed at two different places: on the original image f or on its Max-Tree (which can also be directly interpreted as a pre-processing on the original image). A pre-processing of the input image f generally aims at enhancing the presence of the object of interests in the Max-Tree and/or removing unwanted objects. Then, a pre-processing of the tree may be required in order to ensure that the tree fulfills the hypothesis used by the node classification algorithm.



(a) No filter (b) Filtered tree

Figure 6: Histograms of the attribute circularity before and after Max-Tree filtering (deletion of nodes smaller than 20 pixels).

The main hypothesis of the algorithm is the Gaussian likelihood which is flexible enough to handle most cases. Nevertheless, it is known that the Gaussian model is particularly sensitive to outliers and it is thus important to remove those outliers before running the classification algorithm.

In practice, outliers can be detected by observing the histograms of the observations $\mathbf{Y}$ and searching for features that would hardly fit in a Gaussian mixture model. For example, Fig. 6(a) represents the histogram of the attribute Circularity on a Max-Tree of an image. We see several peaks with large weights that do not fit the Gaussian hypothesis. A careful inspection of the observations through the reconstruction of the spurious nodes allows us to see that those peaks indeed correspond to small nodes where the quantification effects are important: there are many small nodes but only few possible shapes for them.

One simple solution is then to remove the small nodes: this gives a much smoother distribution (see Fig. 6(b)), and the data can be safely injected into the classifier. One can note that the threshold value used to construct Fig. 6(b) is not very sensitive: values between 10 and 30 pixels give similar results. One can also note that the classification algorithm is able to accommodate deviations from the Gaussian model as the truncated distribution at 0 in Fig. 6(b).

Another important question is the selection of the number of classes K. When the dimension of the observation is small (1 or 2), that number can be visually estimated from the histograms by evaluating the number of components needed in a Gaussian mixture model to fit the distribution. In the ideal case, the number of classes K will directly match the number of objects of interest and the association between the twos will be trivial. For example, in the experiment on retinal image segmentation (see Sec. 5.3), we have $K = 2$ with one class for the vessels and the other one for the background. Unfortunately, in some cases K will be larger than the number of objects to identify. For example in an experiment below where we want to separate squares from circles (see Sec. 5.1), we need to introduce a third class to capture the *noise* (*i.e.*, everything that is neither a square nor a circle). Another possible case is that the distribution of the observations among the samples of a single object of interests is multimodal or strongly asymmetric. It is then possible that several classes will be necessary to correctly model this object. In the experiments on retinal image segmentation we have observed that a model with 3 classes (2 for the background and 1 for the vessels) give a better result on some images.

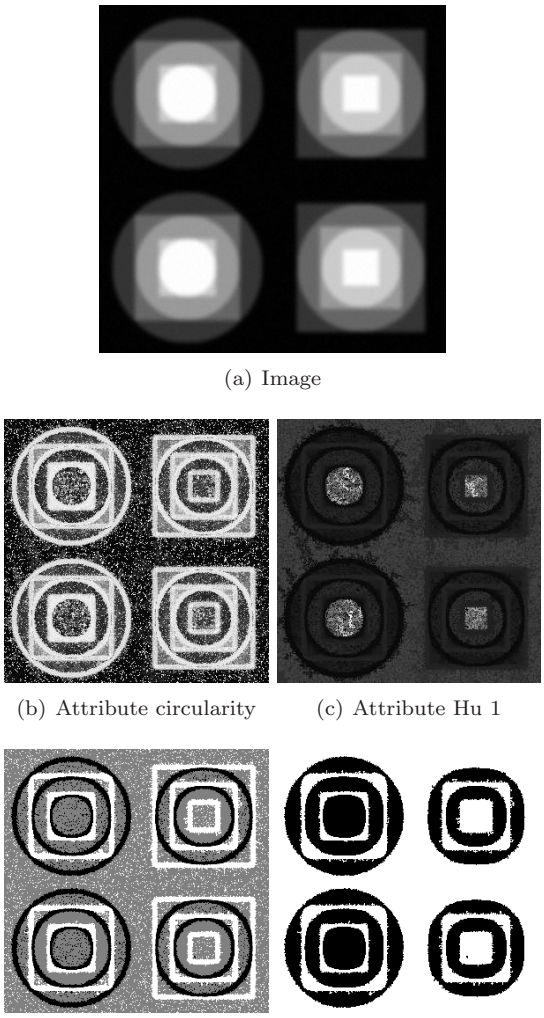
One can note that if K is too large, then the algorithm is subject to numerical instabilities which are easily identified by their two possible manifestations: overflows in numeric values or label switches which prevent the algorithm from converging. The same phenomenon appears in Gaussian mixture model estimation [64].

5. Experiments

In order to evaluate the capacity of the proposed method, we have performed three experiments. The first experiment involves synthetic data and aims at showing the possibility of realizing a shape based classification of the nodes of the Max-Tree in order to separate simple shapes. In a second experiment on astronomical images, we show that our approach enables to reliably detect very faint sources in noisy images and we compare it to two other methods. Finally, in a third experiment on retinal image segmentation, we show that the method can obtain good results on standard benchmarks.

5.1. Multi-scale overlapping shapes classification

This experiment is a demonstration of the method capacity to separate different shapes appearing at various scales, luminosities, and occluding each others. The test image f is represented in Fig. 7(a) and we aim at separating squares from circles.



(a) Image

(b) Attribute circularity (c) Attribute Hu 1

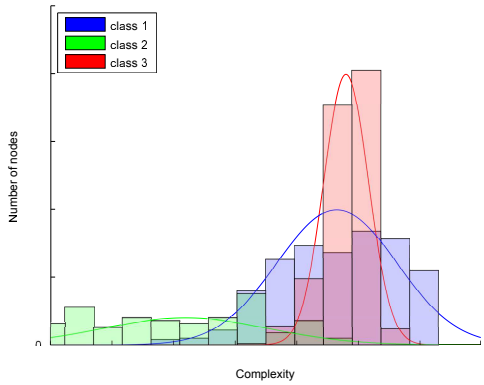
(d) Estimated Labels (e) Filtered Labels

Figure 7: Multi-scale overlapping shapes classification (see text).

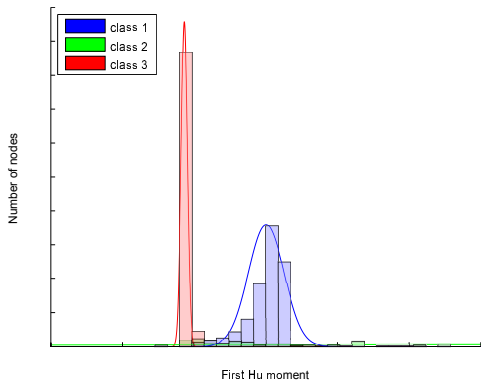
In order to perform this separation we use two classical shapes attributes for the observations: the circularity and the first Hu moment. Formally, let n be a node of the Max-Tree of f, the observation $\mathbf{Y}(n)$ in n is equal to the pair $(\mathrm{Circularity}(n), \mathrm{Hu}_1(n))$. Fig. 7(b) and 7(c) are respectively the reconstruction of the tree with the 2 attribute values $(\mathrm{R}[\mathrm{CC}(f), \mathrm{true}, \mathrm{Circularity}]$, and $\mathrm{R}[\mathrm{CC}(f), \mathrm{true}, \mathrm{Hu}_1])$. The tree is preprocessed by removing the nodes n in $\mathrm{CC}(f)$ smaller than 5 pixels or having a depth smaller than 25: this gives a new set $\mathrm{CC}'(f)$ defined by $\{n \in \mathrm{CC}(f) \mid \mathrm{Area}(n) \geq 5 \text{ and } \mathrm{Depth}(n) \geq 25\}$.

The nodes of the induced by $\mathrm{CC}'(f)$ are classified in 3 classes in order to obtain a first class for circles ω_1, a second class for the noise ω_2, and a third class ω_3 for squares. The reconstruction $\mathrm{R}[\mathrm{CC}'(f), \mathrm{true}, \mathrm{X}]$ of this first classification corresponds to Fig. 7(d). The noise class is filtered out using a simple criterion σ_1 only satisfied if the estimated

node label is different from the label of the noise class ω_2: for each node n in $\mathrm{CC}'(f)$, $\sigma_1(n)$ = true if and only if $\mathrm{X}(n) \neq \omega_2$. The filtered reconstruction $\mathrm{R}[\mathrm{CC}'(f), \sigma_1, \mathrm{X}]$ is shown in Fig. 7(e): circles (in black) and squares (in white) are well separated.



(a) Attribute Circularity



(b) Attribute Hu_1

Figure 8: Empirical distributions (histograms) and estimated distributions (Gaussian curves) of the attributes Circularity and Hu_1 for the 3 estimated classes (respectively in blue, green, and red).

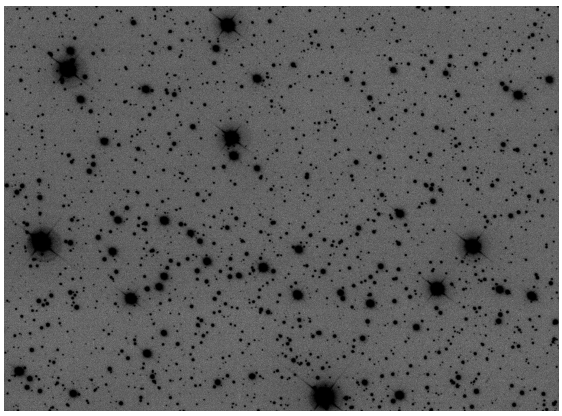
Fig. 8 represents the histograms of the 3 estimated classes for the two attributes – Circularity and Hu_1 – and their estimated probability density functions. It can be seen that:

- the attribute Circularity enables to separate the noise from the two types of shapes, but it has a poor power of separation between squares and circles; and
- the attribute Hu_1 enables to separate squares from circles, but it does not allow us to separate them from noise.

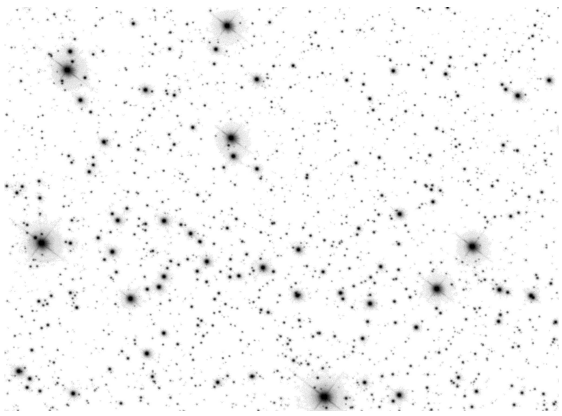
These observations explain the good behavior of the classification process.

5.2. *Astronomical image segmentation and source detection*

In this experiment, we compare our method to a more classical quadtree Markovian segmentation on astronomical images. In such images, the goal is to detect all the sources that shine on a dark background (see Fig. 9(a)). Despite the apparent simplicity of the problem, the difficulty here relies mainly on the large variations in scale (from few pixels to thousands of pixels) and in brightness (from less than 0 dB to dozen of dB) of the various sources. This leads mainly to two difficult cases: the extraction of very faint sources drown in the background noise and the separation of close or superposed sources. One can also note the presence of artifacts that should not be detected as sources, like the diffraction spikes around the bright stars that are caused by the support of the secondary mirror of the telescope.



(a) Astronomical image



(b) Segmentation map

Figure 9: a) Astronomical image from the Sloan Digital Sky Survey (size: 2048×1489 pixels, 32 bits per pixels): the contrast has been inverted and adjusted manually to emphasize faint sources. b) Segmentation map obtained from the Max-Tree Markovian classification.

First, a pre-processing of the original image f is performed in order to compress its dynamic range (astronomical images are stored in linear pixel scale). This gives the

image f' defined by, for any pixel x in E:

$$f'(x) = \frac{f(x)}{|f(x)|} \log(1 + |f(x)|). \tag{7}$$

Then, the Max-Tree of f' is pre-processed by removing very small and very large nodes in order to obtain the set of nodes $\mathrm{CC}'(f') = \{n \in \mathrm{CC}(f') \,|\, 10 \leq \mathrm{Area}(n) \leq 20\,000\}$.

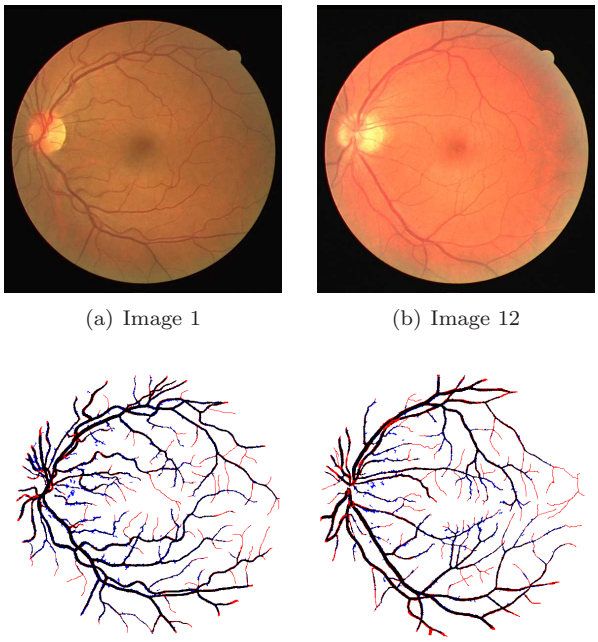
In this application we use the level of the nodes for the observations: for each node n in $\mathrm{CC}'(f')$, $\mathbf{Y}(n)$ is equal to $\mathrm{Level}(n)$. The nodes of the tree induced by $\mathrm{CC}'(f')$ are classified in 9 classes. A segmentation map is obtained by reconstructing the estimated label values $\mathrm{R}[\mathrm{CC}'(f'), \mathrm{true}, \mathrm{X}]$ (see Fig. 9(b), note that labels have been sorted so that largest labels correspond to brightest objects). Finally, in order to obtain a list of sources, we consider that each regional maximum of the segmentation map is a source located on its centroid.

In order to compare our results, we have also run a quadtree Markovian segmentation method – Marsiaa [54][1] – and a reference source extractor software in astronomy – SExtractor[2] – on the same image. Marsiaa was configured to behave as close as possible to our algorithm (Gaussian likelihoods, EM optimization method, and 9 classes) such that the main difference relies in the tree structure used to represent the image. Default parameters were used for SExtractor. Our algorithm, Marsiaa and SExtractor extracted respectively 2157, 1742, and 1799 sources. For comparison, the ninth SDSS Data Release [65] (SDR9[3]) which contains a catalog of sources compiled from several frames taken in several wavelengths and analyzed by several source detection and deblending algorithms contains roughly 3400 sources in the same area of the sky.

Fig. 10(a) is a close up view of Fig. 9(a) with the position of the sources detected by the three algorithms. The segmentation map obtained with our algorithm and Marsiaa are depicted respectively in Fig. 10(b) and 10(c). It can be seen that our algorithm is able to extract very faint sources that were both missed by Marsiaa[54] and SExtractor. Nevertheless, configurations where a faint source is very close to a brighter one is difficult to disambiguate in our scheme as the faint object may not appear as a maximum in the segmentation map: such objects are then missed by the algorithm. A comparison of the two segmentation maps clearly shows the two different tree structures with noisy borders for the Max-Tree and a block effect for the quadtree.

5.3. Retinal image segmentation

In this last experiment, we propose a method to segment the vascular network in retinal images (see for example Fig. 11(a)). The characteristics of this vascular network (length, width, organization...) are markers of several pathologies and its automatic segmentation is an active research field (see [68, 69, 66] among others).



(a) Image 1 (b) Image 12

(c) Segmentation 1 (d) Segmentation 12

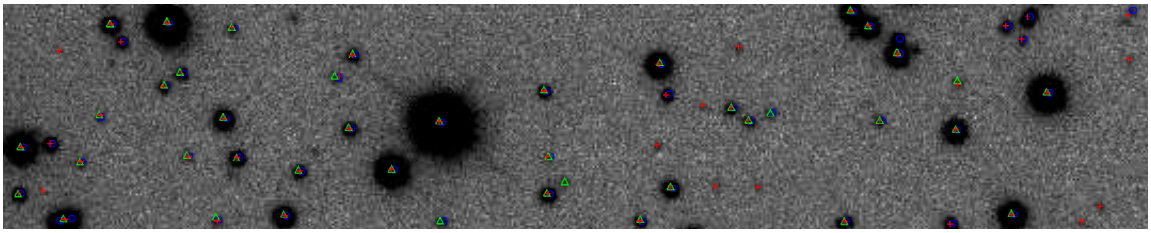
Figure 11: Fig. (a) and (b) are test images 1 and 12 of the DRIVE [66] retinal image benchmark database. Fig. c) and d) are the segmentation results for images (a) and (b) compared to the ground truth. White pixels are true negatives, black pixels are true positives, blue pixels are false positives, and red pixels are false negatives.

In the following, we consider only the green channel of the retinal images. This image is pre-processed using a morphological opening with a flat cross structuring element of 3 pixels diameter in order to remove holes and reconnect some faint vessels. Then, a black top-hat (difference between a closing and the original image) with a circular structuring element of 5 pixels diameter is performed in order to remove large scale variations from the image and to inverse the image: the vascular network now appears as a bright structure over a mostly flat dark background. We denote this image by f'. The Max-Tree of f' is pre-processed by removing very small nodes in order to obtain the new set of nodes $\mathrm{CC}'(f') = \{n \in \mathrm{CC}(f') \,|\, \mathrm{Area}(n) \geq 5\}$.

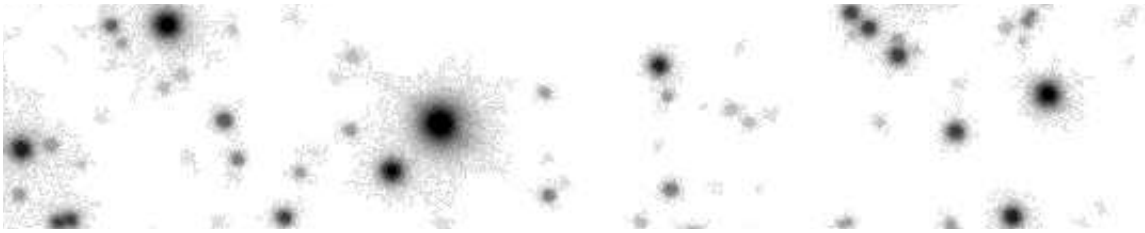
Then we equip each node $n$in $\mathrm{CC}'(f')$ with a vectorial observation composed of the level of the node, the inverse of its first Hu moment and its elongation: *i.e.*, $\mathbf{Y}(n) = (\mathrm{Level}(n), 1/\mathrm{Hu}_1, \mathrm{Elongation}(n))$. The nodes in $\mathrm{CC}'(f')$ are then classified in 2 classes. Finally, a segmentation map is obtained by reconstructing the estimated label values $\mathrm{R}[\mathrm{CC}'(f'), \mathrm{true}, \mathrm{X}]$ (see Fig. 11 (c) an (d) and Fig. 12 (c), (d), (e), and (f)).

This segmentation procedure is evaluated on two standard retinal image databases: Digital Retinal Images for Vessel Extraction (DRIVE [66]) and Structured Analysis of the Retina (STARE [67]). DRIVE is made of 40 color images divided equally in a training set and a test set.

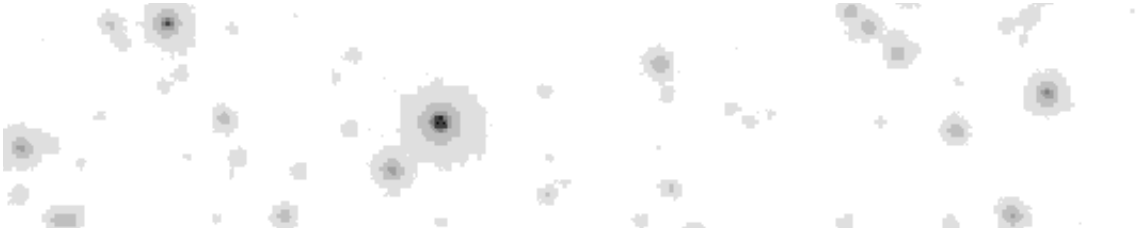
[1] http://lsiit-miv.u-strasbg.fr/paseo/LSBdetection.php#MARSIAA
[2] http://www.astromatic.net/software/sextractor
[3] http://www.sdss3.org/dr9/



(a) Source detection



(b) Segmentation map (proposed algorithm)



(c) Segmentation map (Marsiaa).

Figure 10: Comparison of our algorithm used for source detection and segmentation in astronomical images with Marsiaa and SExtractor. (a) Close up view of Fig. 9(a). Red crosses: sources extracted from the segmentation map (b), green triangles: sources extracted the segmentation map (c), and blue circles: sources extracted by SExtractor. (b) Segmentation map obtained with the Max-Tree Markovian classification. (c) Segmentation map obtained with a quadtree Markovian classification (Marsiaa).

As our method learns directly from each image, the training set is not used. Each of the 20 images of the test set measures 565×584 pixels. It comes with 2 manual segmentations and a mask of the eye fundus (central region of interest). The STARE image database contains 20 color images, all used for tests (no training set). Each image measures 700×605 pixels and comes with 2 manual segmentations. There is no official eye fundus mask for the STARE database but they can be easily derived from the results provided by the authors of the database [67].

Both databases use the same quality measures. For each image, we define:

- the total number of pixels inside the mask of the eye fundus denoted by N;
- the number of True Positives (TP): number of pixels classified as vessel in the segmentation and in the ground truth;
- the number of True Negatives (TN): number of pixels classified as background in the segmentation and in the ground truth (inside the eye fundus mask);
- the True Positive Rate (TPR): the number of true positives (TP) divided by the number of pixels marked as vessel in the ground truth;
- the True Negative Rate (TNR): the number of true negatives (TN) divided by the number of pixels marked as background in the ground truth;
- the Accuracy: the number of true positives (TP) plus the number of true negatives (TN) divided by the total number of pixels N;

In the case of DRIVE, the ground truth corresponds to the first expert segmentation. For STARE, both experts ground truths are used and the results are averaged. Finally the score on each database is defined as the average score obtained on each image of the base. The standard deviation (σ) of the Accuracy is also computed in order to measure the stability of the algorithm.

The results are summarized in Table 1. It can be seen that the method performs well on DRIVE were it places third for the 3 performance measures. The results are more contrasted on STARE where it can be seen that also the method has a quite low TPR it outperforms the other methods in terms of TNR and Accuracy. One can note that among the methods presented in Table 1 most

	DRIVE			STARE		
Method	TPR	TNR	Accuracy (σ)	TPR	TNR	Accuracy (σ)
2nd expert	0.7761	0.9725	0.9473 (0.0048)	0.8949	0.9390	0.9354 (0.0171)
Proposed method	0.7135	0.9778	0.9439 (0.0067)	0.6707	0.9816	0.9514 (0.0104)
Mendonça [69]	0.7344	0.9764	0.9452 (0.0062)	0.6996	0.9730	0.9440 (0.0142)
Staal [66]	0.7193	0.9773	0.9441 (0.0057)	-	-	-
Niemeijer [70]	0.6793	0.9801	0.9416 (0.0065)	-	-	-
Xu [16]	0.6924	0.9779	0.9413 (0.0078)	0.7149	0.9749	0.9471 (0.0114)
Zana [71]	0.6696	0.9769	0.9377 (0.0078)	-	-	-
Al-Diri [68]	-	-	0.9258 (0.0126)	-	-	-
Jiang [72]	0.6478	0.9625	0.9222 (0.0070)	-	-	0.9513
Perez [73]	0.7086	0.9496	0.9181 (0.0240)	-	-	-
Hoover [67]	-	-	-	0.6751	0.9567	0.9267 (0.0099)

Table 1: Segmentation results on the DRIVE and STARE retinal image databases.

are completely dedicated to the retinal vessel network extraction with complicated pre-processings to enhance the presence of faint vessels, while our method is just a simple specialization of the general classification procedure. On the other hand, the method of Xu et al. [16] relies on a morphological filtering of the Max-Tree of the image: the image representation is thus the same as ours but the way it is processed is different. The fact that we obtain quite similar results suggests that both methods seem able to extract the main structures of the Max-Tree with a slightly better sensitivity for our statistical approach.

6. Conclusion

In this article, we have proposed a new way to construct connected filters for image processing and analysis using multivariate attributes. The proposed strategy relies on a hierarchical Markovian classification process. Up-to-our knowledge, this is the first method that is able to handle multivariate attributes in a global and unsupervised way for the construction of connected filters. This approach allowed us to construct robust yet efficient unsupervised algorithms for several image analysis tasks: filtering, segmentation, and source detection. We have shown how to use the method on different experiments involving different types of images, various attributes and several uses of the classification results. Our results were compared to other methods and evaluated on standard image databases.

The benefits of this work can also be seen from the point of view of Markovian segmentation approaches. The use of the Max-Tree as an alternative for the quadtree offers several advantages: better adaptation to the image topology, possibility of using richer attributes as observations, and more compact model.

The possible developments of this works are numerous. Apart from the many possible applications in image analysis, it would be interesting to test the method using others morphological trees, constructed on the image or its gradient, like the binary partition tree [34] or the tree of shapes [37, 38]. From the point of view of the classification method, several developments are desirable: the automatic optimization of the number of classes [74], the use of more general multivariate probability density functions for the modelization of the data driven term (like copula)[75], or the automatic selection of the most relevant attributes for a given problem.

References

[1] C. Lantuéjoul, S. Beucher, On the use of the geodesic metric in image analysis, Journal of Microscopy 121 (1980) 29–49. doi:10.1111/j.1365-2818.1981.tb01197.x.

[2] C. Lantuéjoul, F. Maisonneuve, Geodesic methods in quantitative image analysis, Pattern Recognition 17 (1984) 177–187. doi:10.1016/0031-3203(84)90057-8.

[3] J. Crespo, R. Schafer, Locality and adjacency stability constraints for morphological connected operators, Journal of Mathematical Imaging and Vision 7 (1) (1997) 85–102. doi:10.1023/A:1008270125009.

[4] H. Heijmans, Connected morphological operators for binary images, Computer Vision and Image Understanding 73 (1) (1999) 99–120. doi:10.1006/cviu.1998.0703.

[5] J. Serra, P. Salembier, Connected operators and pyramids, in: E. R. Dougherty, P. D. Gader, J. C. Serra (Eds.), Image Algebra and Morphological Image Processing IV, Vol. 2030, SPIE, 1993, pp. 65–76. doi:10.1117/12.146672.

[6] P. Salembier, J. Serra, Flat zones filtering, connected operators, and filters by reconstruction, IEEE Transactions on Image Processing 4 (8) (1995) 1153–1160. doi:10.1109/83.403422.

[7] E. Breen, R. Jones, Attribute openings, thinnings, and granulometries, Computer Vision and Image Understanding 64 (3) (1996) 377–389. doi:10.1006/cviu.1996.0066.

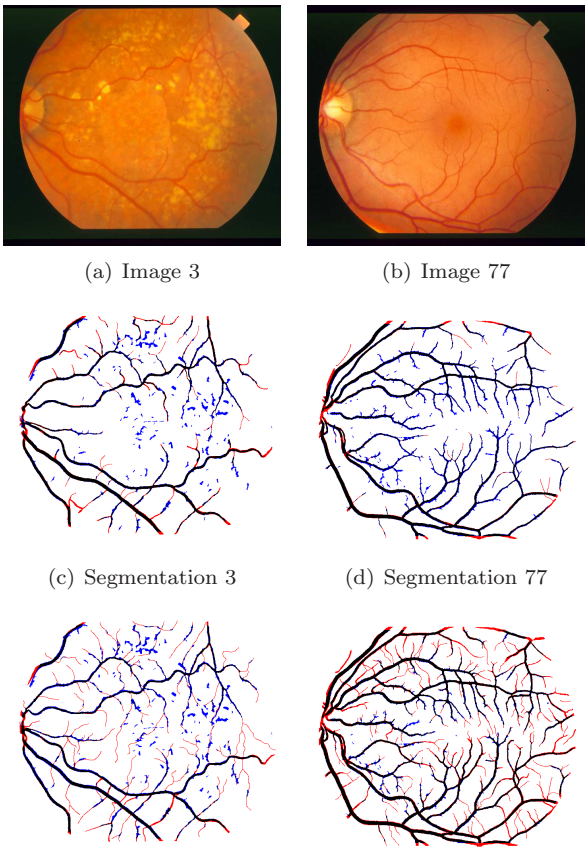
[8] R. Jones, Connected filtering and segmentation using component trees, Computer Vision and Image Understanding 75 (3) (1999) 215–228. doi:10.1006/cviu.1999.0777.

[9] P. Salembier, A. Oliveras, L. Garrido, Anti-extensive connected operators for image and sequence processing, IEEE Transactions on Image Processing 7 (4) (1998) 555–570. doi:10.1109/83.663500.

[10] B. Naegel, L. Wendling, A document binarization method based on connected operators, Pattern Recognition Letters 31 (11) (2010) 1251–1259. doi:10.1016/j.patrec.2010.04.003.

[11] G. Ouzounis, M. Wilkinson, Hyperconnected attribute filters based on k-flat zones, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2) (2011) 224–239. doi:10.1109/TPAMI.2010.74.

[12] B. Perret, S. Lefèvre, C. Collet, E. Slezak, Hyperconnections and hierarchical representations for grayscale and multiband im-



(a) Image 3 (b) Image 77

(c) Segmentation 3 (d) Segmentation 77

(e) Segmentation 3 (f) Segmentation 77

Figure 12: Fig. (a) and (b) are test images 3 and 77 of the STARE [67] retinal image benchmark database. Fig. (c) and (d) (resp. (e) and (f)) are the segmentation results for images (a) and (b) compared to the first (resp. second) ground truth of the base. White pixels are true negatives, black pixels are true positives, blue pixels are false positives, and red pixels are false negatives.

age processing, IEEE Transactions on Image Processing 21 (1) (2012) 14–27. doi:10.1109/TIP.2011.2161322.

[13] M. Wilkinson, M. Westenberg, Shape preserving filament enhancement filtering, in: W. Niessen, M. Viergever (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001, Vol. 2208 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2001, pp. 770–777. doi:10.1007/3-540-45468-3_92.

[14] G. Ouzounis, M. Wilkinson, Mask-based second-generation connectivity and attribute filters, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 990–1004. doi:10.1109/TPAMI.2007.1045.

[15] A. Dufour, O. Tankyevych, B. Naegel, H. Talbot, C. Ronse, J. Baruthio, P. Dokládal, N. Passat, Filtering and segmentation of 3d angiographic data: Advances based on mathematical morphology, Medical Image Analysis 17 (2) (2013) 147–164. doi:10.1016/j.media.2012.08.004.

[16] Y. Xu, T. Géraud, L. Najman, Morphological filtering in shape spaces: Applications using tree-based image representations, in: International Conference on Pattern Recognition (ICPR), 2012 21st, 2012, pp. 485–488.

[17] C. Kurtz, N. Passat, P. Gançarski, A. Puissant, Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology, Pattern Recognition 45 (2) (2012) 685 – 706. doi:http://dx.doi.org/10.1016/j.patcog.2011.07.017.

[18] A. Alonso-González, S. Valero, J. Chanussot, C. López-Martínez, P. Salembier, Processing multidimensional sar and hyperspectral images with binary partition tree, Proceedings of the IEEE 101 (3) (2013) 723–747. doi:10.1109/JPROC.2012.2205209.

[19] C. Berger, T. Géraud, R. Levillain, N. Widynski, A. Baillard, E. Bertin, Effective component tree computation with application to pattern recognition in astronomical imaging, in: IEEE International Conference on Image Processing, Vol. 4, 2007, pp. 41–44. doi:10.1109/ICIP.2007.4379949.

[20] B. Perret, S. Lefèvre, C. Collet, E. Slezak, Connected component trees for multivariate image processing and applications in astronomy, in: IAPR, International Conference on Pattern Recognition, 2010, pp. 4089–4092. doi:DOI:10.1109/ICPR.2010.994.

[21] M. Westenberg, J. Roerdink, M. Wilkinson, Volumetric attribute filtering and interactive visualization using the max-tree representation, IEEE Transactions on Image Processing 16 (12) (2007) 2943–2952.

[22] N. Passat, B. Naegel, F. Rousseau, M. Koob, J.-L. Dietemann, Interactive segmentation based on component-trees, Pattern Recognition 44 (10–11) (2011) 2539–2554. doi:10.1016/j.patcog.2011.03.025.

[23] L. Chen, M. W. Berry, W. W. Hargrove, Using dendronal signatures for feature extraction and retrieval, International Journal of Imaging Systems and Technology 11 (4) (2000) 243–253. doi:10.1002/ima.1009.

[24] E. R. Urbach, J. B. T. M. Roerdink, M. H. F. Wilkinson, Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2) (2007) 272–285. doi:10.1109/TPAMI.2007.28.

[25] J. Mattes, M. Richard, J. Demongeot, Tree representation for image matching and object recognition, in: G. Bertrand, M. Couprie, L. Perroton (Eds.), Discrete Geometry for Computer Imagery, Vol. 1568 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1999, pp. 298–309. doi:10.1007/3-540-49126-0_23.

[26] A. Falcão, J. Stolfi, R. de Alencar Lotufo, The image foresting transform: Theory, algorithms, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (1) (2004) 19–29. doi:10.1109/TPAMI.2004.10012.

[27] J. Darbon, M. Sigelle, Image restoration with discrete constrained total variation part i: Fast and exact optimization, Journal of Mathematical Imaging and Vision 26 (3) (2006) 261–276. doi:10.1007/s10851-006-8803-0.

[28] J. Serra, Mathematical morphology for boolean lattices, in: J. Serra (Ed.), Image Analysis and Mathematical Morphology. II: Theoretical Advances, Academic Press, 1988, pp. 37–58.

[29] C. Ronse, Set-theoretical algebraic approaches to connectivity in continuous or digital spaces, Journal of Mathematical Imaging and Vision 8 (1) (1998) 41–58. doi:10.1023/A:1008210216583.

[30] J. Serra, Connectivity on complete lattices, Journal of Mathematical Imaging and Vision 9 (3) (1998) 231–251. doi:10.1007/978-1-4613-0469-2_11.

[31] N. Passat, B. Naegel, Component-hypertrees for image segmentation, in: P. Soille, M. Pesaresi, G. K. Ouzounis (Eds.), Mathematical Morphology and Its Applications to Image and Signal Processing, Vol. 6671 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 284–295. doi:10.1007/978-3-642-21569-8_25.

[32] B. Perret, J. Cousty, O. Tankyevych, H. Talbot, N. Passat, Directed connected operators: asymmetric hierarchies for image filtering Tech. rep., Laboratoire d'Informatique Gaspard-Monge - LIGM, Université Paris Est (2013).
URL http://hal.archives-ouvertes.fr/hal-00869727

[33] B. Perret, Inf-structuring functions: A unifying theory of connections and connected operators, Journal of Mathematical Imaging and Vision (2014) 1–24doi:10.1007/s10851-014-0515-2.

[34] P. Salembier, L. Garrido, Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval, IEEE Transactions on Image Processing 9 (4) (2000) 561–576. doi:10.1109/83.841934.

[35] P. Soille, Constrained connectivity for hierarchical image partitioning and simplification, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (7) (2008) 1132–1145. doi:10.1109/TPAMI.2007.70817.

[36] J. Cousty, L. Najman, Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts, in: P. Soille, M. Pesaresi, G. Ouzounis (Eds.), Mathematical Morphology and Its Applications to Image and Signal Processing, Vol. 6671 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 272–283. doi:10.1007/978-3-642-21569-8_24.

[37] P. Monasse, F. Guichard, Scale-space from a level lines tree, Journal of Visual Communication and Image Representation 11 (2) (2000) 224–236. doi:10.1006/jvci.1999.0441.

[38] T. Géraud, E. Carlinet, S. Crozet, L. Najman, A quasi-linear algorithm to compute the tree of shapes of nd images, in: C. Hendriks, G. Borgefors, R. Strand (Eds.), Mathematical Morphology and Its Applications to Signal and Image Processing, Vol. 7883 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 98–110. doi:10.1007/978-3-642-38294-9_9.

[39] C. Tzafestas, P. Maragos, Shape connectivity: Multiscale analysis and application to generalized granulometries, Journal of Mathematical Imaging and Vision 17 (2) (2002) 109–129. doi:10.1023/A:1020629402912.

[40] G. Ouzounis, P. Soille, The alpha-tree algorithm., Tech. rep., Publications Office of the European Union (2012). doi:10.2788/48773.

[41] J. Cousty, L. Najman, B. Perret, Constructive links between some morphological hierarchies on edge-weighted graphs, in: C. L. L. Hendriks, G. Borgefors, R. Strand (Eds.), Mathematical Morphology and Its Applications to Signal and Image Processing, Vol. 7883 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 86–97. doi:10.1007/978-3-642-38294-9_8.

[42] L. Najman, T. Géraud, Discrete set-valued continuity and interpolation, in: C. Hendriks, G. Borgefors, R. Strand (Eds.), Mathematical Morphology and Its Applications to Signal and Image Processing, Vol. 7883 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 37–48. doi:10.1007/978-3-642-38294-9_4.

[43] L. Vincent, Grayscale area openings and closings, their efficient implementation and applications, in: EURASIP First Workshop on Mathematical Morphology and its Applications to Signal Processing, 1993, pp. 22–27.

[44] L. Vincent, Morphological area openings and closings for greyscale images, in: O. Ying-Lie, A. Toet, D. Foster, H. Heijmans, P. Meer (Eds.), Shape in Picture, Vol. 126 of NATO ASI Series, Springer Berlin Heidelberg, 1994, pp. 197–208. doi:10.1007/978-3-662-03039-4_13.

[45] M. Grimaud, New measure of contrast: the dynamics, in: P. D. Gader, E. R. Dougherty, J. C. Serra (Eds.), Image Algebra and Morphological Image Processing III, Vol. 1769 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 1992, pp. 292–305.

[46] Y. Xu, T. Géraud, L. Najman, Salient Level Lines Selection Using the Mumford-Shah Functional, in: Internation Conference on Image Processing 2013, Melbourne, Australie, 2013, pp. 1–5.

[47] Y. Xu, T. Géraud, L. Najman, Two applications of shape-based morphology: Blood vessels segmentation and a generalization of constrained connectivity, in: C. Hendriks, G. Borgefors, R. Strand (Eds.), Mathematical Morphology and Its Applications to Signal and Image Processing, Vol. 7883 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 390–401. doi:10.1007/978-3-642-38294-9_33.

[48] E. Urbach, N. Boersma, M. Wilkinson, Vector-attribute filters, in: C. Ronse, L. Najman, E. Decencière (Eds.), Mathematical Morphology: 40 Years On, Vol. 30 of Computational Imaging and Vision, Springer Netherlands, 2005, pp. 95–104. doi:10.1007/1-4020-3443-1_10.

[49] B. Naegel, N. Passat, N. Boch, M. Kocher, Segmentation using vector-attribute filters: methodology and application to dermatological imaging, in: G. Bannon, J. Barrera, U. Braga-Neto (Eds.), International Symposium on Mathematical Morphology, Vol. 1, Rio de Janeiro, Brazil, 2007, pp. 239–250.

[50] C. Bouman, M. Shapiro, A multiscale random field model for bayesian image segmentation, IEEE Transactions on Image Processing 3 (1994) 162–177. doi:10.1109/83.277898.

[51] S. Geman, D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (6) (1984) 721–741. doi:10.1109/TPAMI.1984.4767596.

[52] J. Whittaker, Graphical models in applied multivariate statistics, Wiley series in probability and mathematical statistics: Probability and mathematical statistics, Wiley, 1990.

[53] J.-M. Laferté, P. Pérez, F. Heitz, Discrete markov image modeling and inference on the quadtree, IEEE Transactions on Image Processing 9 (3) (2000) 390–404. doi:10.1109/83.826777.

[54] J.-N. Provost, C. Collet, P. Rostaing, P. Pérez, P. Bouthemy, Hierarchical Markovian segmentation of multispectral images for the reconstruction of water depth maps, Computer Vision and Image Understanding 93 (2) (2004) 155–174. doi:10.1016/j.cviu.2003.07.004.

[55] S. Le Cam, C. Collet, F. Salzenstein, Detection of transient signals in lung sounds: Local approach using a markovian tree with frequency selectivity, Journal of Signal Processing Systems 65 (3) (2011) 445–456. doi:10.1007/s11265-010-0529-y.

[56] W. Pieczynski, Pairwise markov chains, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 634–639. doi:10.1109/TPAMI.2003.1195998.

[57] W. Pieczynski, Triplet markov chains and image segmentation, in: A. Mohammed-Djafari (Ed.), Inverse problems in Vision and 3D Tomography, Wiley, 2013, pp. 123–153. doi:10.1002/9781118603864.ch4.

[58] E. Monfrini, J. Lecomte, F. Desbouvries, W. Pieczynski, Image and signal restoration using pairwise markov trees, in: IEEE Workshop on Statistical Signal Processing, St Louis, Missouri, USA, 2003, pp. 174–177. doi:10.1109/SSP.2003.1289372.

[59] F. Salzenstein, C. Collet, Fuzzy markov random fields versus chains for multispectral image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (11) (2006) 1753–1767. doi:10.1109/TPAMI.2006.228.

[60] L. Najman, M. Couprie, Building the component tree in quasi-linear time, IEEE Transactions on Image Processing 15 (11) (2006) 3531–3539. doi:10.1109/TIP.2006.877518.

[61] E. Carlinet, T. Géraud, A comparison of many max-tree computation algorithms, in: C. Hendriks, G. Borgefors, R. Strand (Eds.), Mathematical Morphology and Its Applications to Signal and Image Processing, Vol. 7883 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 73–85. doi:10.1007/978-3-642-38294-9_7.

[62] M.-K. Hu, Visual pattern recognition by moment invariants, IRE Transactions on Information Theory 8 (2) (1962) 179–187. doi:10.1109/TIT.1962.1057692.

[63] J. Pearl, Reverend bayes on inference engines: a distributed hierarchical approach, in: Proceedings of the National Conference on Artificial Intelligence, 1982, pp. 133–136.

[64] J. Ghosh, P. Sen, On the Asymptotic Performance of the Log Likelihood Ratio Statistic for the Mixture Model and Related Results, Institute of Statistics mimeo series, University of North Carolina, 1984.

[65] C. P. Ahn, R. Alexandroff, C. Allende Prieto, S. F. Anderson, T. Anderton, B. H. Andrews, É. Aubourg, S. Bailey, E. Balbinot, R. Barnes, et al., The Ninth Data Release of the Sloan Digital Sky Survey: First Spectroscopic Data from the SDSS-III Baryon Oscillation Spectroscopic Survey, The Astrophysical Journal Supplement 203 (2012) 21. doi:10.1088/0067-0049/203/2/21.

[66] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, B. van Gin-

neken, Ridge-based vessel segmentation in color images of the retina, IEEE Transactions on Medical Imaging 23 (4) (2004) 501–509. doi:10.1109/TMI.2004.825627.

[67] A. Hoover, V. Kouznetsova, M. Goldbaum, Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response, IEEE Transactions on Medical Imaging 19 (3) (2000) 203–210. doi:10.1109/42.845178.

[68] B. Al-Diri, A. Hunter, D. Steel, An active contour model for segmenting and measuring retinal vessels, IEEE Transactions on Medical Imaging 28 (9) (2009) 1488–1497. doi:10.1109/TMI.2009.2017941.

[69] A. Mendonça, A. Campilho, Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction, IEEE Transactions on Medical Imaging 25 (9) (2006) 1200–1213. doi:10.1109/TMI.2006.879955.

[70] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, M. Abramoff, Comparative study of retinal vessel segmentation methods on a new publicly available database, in: J. Michael Fitzpatrick, M. Sonka (Eds.), SPIE Medical Imaging, Vol. 5370 of SPIE, 2004, pp. 648–656.

[71] F. Zana, J.-C. Klein, Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation, IEEE Transactions on Image Processing 10 (7) (2001) 1010–1019. doi:10.1109/83.931095.

[72] X. Jiang, D. Mojon, Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (1) (2003) 131–137. doi:10.1109/TPAMI.2003.1159954.

[73] M. Martínez-Pérez, A. Hughes, A. Stanton, S. Thom, A. Bharath, K. Parker, Retinal blood vessel segmentation by means of scale-space analysis and region growing, in: C. Taylor, A. Colchester (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI'99, Vol. 1679 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1999, pp. 90–97. doi:10.1007/10704282_10.

[74] C. Collet, F. Murtagh, Multiband segmentation based on a hierarchical markov model, Pattern Recognition 37 (12) (2004) 2337–2347. doi:10.1016/j.patcog.2004.03.017.

[75] F. Flitti, C. Collet, A. Joannic-Chardin, Unsupervised multiband image segmentation using hidden markov quadtree and copulas, in: IEEE International Conference on Image Processing, 2005., Vol. 2, 2005, pp. II–634–7. doi:10.1109/ICIP.2005.1530135.

AppendixA. Algorithms

Algorithm 1: Bottom-Up/Top-Down tree segmentation.

Input: The Max-Tree of the function f.
Input: The observation $\mathbf{Y}(n)$ on $\mathrm{CC}(f)$.
Input: Set of labels Ω_X.
Input: Model transition probabilities $t_{kk'}$ (5).
Input: Model root probabilities π_k (6).
Input: Classes mean values $\mu_{\mathbf{k}}$ and covariances matrices $\mathbf{\Sigma_k}$ (4).
Output: The label $\mathrm{X}(n)$ on $\mathrm{CC}(f)$.

```
/* Computation of priors */
```

for *each node* n *in* $\mathrm{CC}(f)$ **from** *the root* **to** *the leaves* **do**
  if n *is not the root* **then**
    for *each label* i *in* Ω_X **do**

$$P(\mathrm{X}(n) = i) \leftarrow \sum_{j \in \Omega_X} P(\mathrm{X}(n^-) = j) t_{ji};$$

```
/* Top-Down pass */
```

for *each node* n *in* $\mathrm{CC}(f)$ **from** *the leaves* **to** *the root* **do**
  for *each label* i *in* Ω_X **do**

$$P(\mathrm{X}(n) = i \mid \mathbf{Y}(n >)) \leftarrow \frac{1}{Z} P(\mathrm{X}(n) = i) \prod_{c \in n^+} \sum_{j \in \Omega_X} \frac{P(\mathrm{X}(n) = i) P(\mathrm{X}(c) = j \mid \mathbf{Y}(c >)) t_{ij}}{P(\mathrm{X}(c) = j)};$$

/* Z is a normalization factor such that $\sum_{i \in \Omega_X} P(\mathrm{X}(n) = i \mid \mathbf{Y}(n >)) = 1$ */

/* Note that for the root node r we have $P(\mathrm{X}(r) \mid \mathbf{Y}(r >)) = P(\mathrm{X}(r) \mid \mathbf{Y})$ */

```
/* Bottom-Up pass */
```

for *each node* n *in* $\mathrm{CC}(f)$ **from** *the leaves* **to** *the root* **do**
  if n *is not the root* **then**
    for *each label* i *in* Ω_X **do**
      for *each label* j *in* Ω_X **do**

$$P(\mathrm{X}(n) = i, \mathrm{X}(n^-) = j \mid \mathbf{Y}) \leftarrow P(\mathrm{X}(n^-) = j \mid \mathbf{Y}) \frac{P(\mathrm{X}(n) = i \mid \mathbf{Y}(n >)) P(\mathrm{X}(n^-) = j) t_{ji} / P(\mathrm{X}(n) = i)}{\sum_{\ell \in \Omega_X} P(\mathrm{X}(n) = \ell \mid \mathbf{Y}(n >)) P(\mathrm{X}(n^-) = j) t_{j\ell} / P(\mathrm{X}(n) = \ell)}$$

$$P(\mathrm{X}(n) = i \mid \mathbf{Y}) = \sum_{j \in \Omega_X} P(\mathrm{X}(n) = i, \mathrm{X}(n^-) = j \mid \mathbf{Y});$$

$$\mathrm{X}(n) \leftarrow \arg\max_{i \in \Omega_X} P(\mathrm{X}(n) = i \mid \mathbf{Y})$$

Algorithm 2: Unsupervised EM/Markovian classification of the node of the tree.

Input: CC(f), a Max-Tree of the function f.
Input: The observation $\mathbf{Y}(n)$ for each node in CC(f).
Input: Set of K labels Ω_X.
Output: The label $\mathrm{X}(n)$ of each node n in CC(f).

```
/* Initialization                                                                    */
```

$gmm \leftarrow$ estimation on $\mathbf{Y}$ of a Gaussian Mixture Model with K components;
for *each label i in Ω_X* **do**

```
    /* All labels are equipossible at the root node.                                 */
```

  $\pi_i \leftarrow \frac{1}{K}$;

```
    /* There is a high probability that a child has the same label as its father.    */
```

  $t_{ii} \leftarrow 0.75$;
  for *each label $j \neq i$ in Ω_X* **do**
    $t_{ij} \leftarrow \frac{0.25}{K-1}$;
  $\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i \leftarrow$ mean and covariance matrix of the i-th component of gmm;

```
/* Iterative unsupervised segmentation                                               */
```

do
  Bottom-Up/Top-Down tree segmentation (Algorithm 1);

```
  /* Model update with EM                                                            */
```

  $r \leftarrow$ root node of CC(f);
  for *each label i in Ω_X* **do**
    $\pi_i \leftarrow P(\mathrm{X}(r) = i \mid \mathbf{Y})$;
    for *each label $j \neq i$ in Ω_X* **do**

$$t_{ij} \leftarrow \frac{\sum\limits_{\substack{n \in \mathrm{CC}(f) \\ n \neq r}} P(\mathrm{X}(n) = j, \mathrm{X}(n^-) = i \mid \mathbf{Y})}{\sum\limits_{\substack{n \in \mathrm{CC}(f) \\ n \neq r}} P(\mathrm{X}(n^-) = i \mid \mathbf{Y})};$$

$$\boldsymbol{\mu}_i \leftarrow \frac{\sum\limits_{n \in \mathrm{CC}(f)} \mathbf{Y}(n) P(\mathrm{X}(n) = i \mid \mathbf{Y})}{\sum\limits_{n \in \mathrm{CC}(f)} P(\mathrm{X}(n) = i \mid \mathbf{Y})};$$

$$\boldsymbol{\Sigma}_i \leftarrow \frac{\sum\limits_{n \in \mathrm{CC}(f)} P(\mathrm{X}(n) = i \mid \mathbf{Y})(\mathbf{Y}(n) - \boldsymbol{\mu}_i)(\mathbf{Y}(n) - \boldsymbol{\mu}_i)^T}{\sum\limits_{n \in \mathrm{CC}(f)} P(\mathrm{X}(n) = i \mid \mathbf{Y})};$$

until *less than 3% of the labels have changed between two iterations*;