Pierre Soille
Martino Pesaresi
Georgios K. Ouzounis (Eds.)

LNCS 6671

Mathematical Morphology and Its Applications to Image and Signal Processing

10th International Symposium, ISMM 2011
Verbania-Intra, Italy, July 2011
Proceedings

Springer

Pierre Soille Martino Pesaresi
Georgios K. Ouzounis (Eds.)

Mathematical Morphology and Its Applications to Image and Signal Processing

10th International Symposium, ISMM 2011
Verbania-Intra, Italy, July 6-8, 2011
Proceedings

Springer

Volume Editors

Pierre Soille
Martino Pesaresi
Georgios K. Ouzounis
European Commission, Joint Research Centre
Institute for the Protection and Security of the Citizen
via Enrico Fermi, 2749, 21027, Ispra (VA), Italy
E-mail: {pierre.soille, martino.pesaresi, georgios.ouzounis}@jrc.ec.europa.eu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21568-1 e-ISBN 978-3-642-21569-8
DOI 10.1007/978-3-642-21569-8
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011928710

CR Subject Classification (1998): I.4.10, I.5.4, C.3, J.3, G.2, I.2.8

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This LNCS volume on Mathematical Morphology and Its Applications to Image and Signal Processing contains the full papers accepted for presentation at the 10th International Symposium on Mathematical Morphology (ISMM 2011), held in Intra, Italy, 6th–8th of July, 2011. ISMM is a biannual event bringing together researchers, students, and practitioners of Mathematical Morphology to present and discuss advances on topics ranging from new theoretical developments to novel applications, solving complex image analysis problems. ISMM was established as the main scientific event in the field and this anniversary edition marked the tenth successful organisation in the series that was initiated in 1993 in Barcelona.

The call for papers was answered with 49 submissions. Each submitted paper was peer-reviewed by three referees selected from the Programme Committee. Based on their reviews, a total of 39 papers were accepted for publication in this volume, 27 of which were selected for oral, and 12 for poster presentation. The final programme of ISMM 2011 was divided into nine thematic areas: theory, lattices and order, connectivity, image analysis, processing and segmentation, adaptive morphology, algorithms, remote sensing, visualisation, and applications.

The topic of special attention for ISMM 2011 was the adaptation of morphological methods for the analysis of geo-spatial data. It shaped the separate section on remote sensing consisting of five contributions, further backed by four papers on the topics of connectivity and algorithms. The symposium programme was enriched by the following three keynote lectures:

– *"Applications of Discrete Calculus in Computational Science"* by Leo Grady (Siemens Corporate Research);
– *"Morphological Profiles in Classification of Remote Sensing Imagery"* by Jón Atli Benediktsson (University of Iceland);
– *"Mathematical Morphology in Computer Graphics, Scientific Visualization and Visual Exploration"* by Jos B.T.M. Roerdink (University of Groningen).

ISMM 2011 was organised by the Institute for the Protection and Security of the Citizen (IPSC), of the European Commission's Joint Research Centre in Ispra, Italy. The success of the event is attributed to the joint effort of many individuals. In particular, we wish to thank all the authors who accepted our invitation, the members of the Programme Committee for delivering thorough reviews of the submitted manuscripts, the invited speakers for offering three high-quality lectures, and the Session Chairs for running the symposium smoothly. We wish to acknowledge the IPSC (JRC) for supporting this event, and the hotel Il Chiostro, Intra, for facilitating the symposium and taking care of the organisation of the social event. Special thanks go to Ana-Maria Duta (IPSC) for assisting in organisational issues. All submitted material was managed by the online EasyChair conference management system.

April 2011
Ispra, Italy

P. Soille, M. Pesaresi, and G.K. Ouzounis
EC Joint Research Centre

ISMM 2011 Organisation

ISMM 2011 was organised by the Geo-spatial Information Analysis for Security and Stability Action of the Global Security and Crisis Management Unit, Institute for the Protection and Security of the Citizen, Joint Research Centre, European Commission.

Chairing Committee

Pierre Soille	EC Joint Research Centre, Ispra, Italy
Martino Pesaresi	EC Joint Research Centre, Ispra, Italy
Georgios K. Ouzounis	EC Joint Research Centre, Ispra, Italy

Steering Committee

Junior Barrera	Universidade de São Paolo, Brazil
Isabelle Bloch	Télécom ParisTech (ENST), France
Renato Keshet	HP Labs, Haifa, Israel
Petros Maragos	National Technical University of Athens, Greece
Fernand Meyer	Mines ParisTech, Fontainebleau, France
Jos B.T.M. Roerdink	University of Groningen, The Netherlands
Christian Ronse	Université de Strasbourg, France
Philippe Salembier	Universita Polytècnica de Catalunya, Barcelona, Spain
Pierre Soille	EC Joint Research Centre, Ispra, Italy
Hugues Talbot	Université Paris-Est, ESIEE, France
Michael H.F. Wilkinson	University of Groningen, The Netherlands

Local Organising Committee

Ana-Maria Duta
Georgios K. Ouzounis
Martino Pesaresi
Pierre Soille

Invited Speakers

Jón Atli Benediktsson	University of Iceland, Reykjavik, Iceland
Leo Grady	Siemens Corporate Research, Princeton, USA
Jos B.T.M. Roerdink	University of Groningen, The Netherlands

Programme Committee

Jesús Angulo
Akira Asano
Isabelle Bloch
Gunilla Borgefors
Michael Buckley
Bernhard Burgeth
Jocelyn Chanussot
Jean Cousty
José Crespo
Etienne Decencière
Adrian Evans
Jacopo Grazzini
Thierry Géraud
Lionel Gueguen
Allan Hanbury
Marcin Iwanowski
Andrei Jalba
Dominique Jeulin
Renato Keshet
Ullrich Köthe
Sébastien Lefèvre
Roberto Lotufo
Cris L. Luengo Hendriks
Beatriz Marcotegui
Fernand Meyer
Annick Montanvert
Laurent Najman
Nicolas Passat
Jos B.T.M. Roerdink
Christian Ronse
Philippe Salembier
Gabriella Sanniti di Baja
Jean Serra
Hugues Talbot
Iván Ramon Terol-Villalobos
Erik Urbach
Marc Van Droogenbroeck
Michel Westenberg
Michael Wilkinson

Additional Reviewers

Pascal Gwosdek
André Körbes
Benjamin Perret
Letícia Rittner
Oliver Vogel

Table of Contents

Theory

Lattices and Order

Connectivity

Image Analysis, Processing, and Segmentation

Adaptive Morphology

Algorithms

Remote Sensing

Visualisation

Applications

Sparse Mathematical Morphology Using Non-negative Matrix Factorization

Jesús Angulo and Santiago Velasco-Forero

CMM-Centre de Morphologie Mathématique, Mathématiques et Systèmes, MINES
ParisTech; 35, rue Saint Honoré, 77305 Fontainebleau Cedex, France
{jesus.angulo,santiago.velasco}@mines-paristech.fr

Abstract. Sparse modelling involves constructing a succinct representation of initial data as a linear combination of a few typical atoms of a dictionary. This paper deals with the use of sparse representations to introduce new nonlinear operators which efficiently approximate the dilation/erosion. Non-negative matrix factorization (NMF) is a dimensional reduction (i.e., dictionary learning) paradigm particularly adapted to the nature of morphological processing. Sparse NMF representations are studied to introduce pseudo-morphological binary dilations/erosions. The basic idea consists in processing exclusively the image dictionary and then, the result of processing each image is approximated by multiplying the processed dictionary by the coefficient weights of the current image. These operators are then extended to grey-level images by means of the level-set decomposition. The performance of the present method is illustrated using families of binary shapes and face images.

1 Introduction

Mathematical morphology [11,4] is a nonlinear image processing methodology based on the application of lattice theory to spatial structures. Morphological filters and transformations are useful for various image processing tasks [12], such as denoising, contrast enhancement, multi-scale decomposition, feature extraction and object segmentation. In addition, morphological operators are defined using very intuitive geometrical notions which allows us the perceptual development and interpretation of complex algorithms by combination of various operators.

Let E be a space of points, which is considered here as a finite digital space of the pixels of the image, i.e., $E \subset \mathbf{Z}^2$ such that $N = |E|$ is the number of pixels. Image intensities are numerical values, which ranges in a closed subset $\mathcal{T}$ of $\overline{\mathbf{R}} = \mathbf{R} \cup \{-\infty, +\infty\}$; for example, for an image of discrete L values, it can be assumed $\mathcal{T} = \{t_1, t_2, \cdots, t_L\}$. Then, a binary image X is modelled as a subset of E, i.e., $X \in \mathcal{P}(E)$; a grey-level image $f(p_i)$, where $p_i \in E$ are the pixel coordinates, is a numerical function $E \to \mathcal{T}$, i.e., $f \in \mathcal{F}(E, \mathcal{T})$. In mathematical morphology, an operator ψ is a map transforming an image into an image. There are thus operators on binary images, i.e., maps $\mathcal{P}(E) \to \mathcal{P}(E)$; or on grey-level images, i.e., maps $\mathcal{F}(E, \mathcal{T}) \to \mathcal{F}(E, \mathcal{T})$.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 1–12, 2011.

Sparse coding and dictionary learning, where data is assumed to be well represented as a linear combination of a few elements from a dictionary, is an active research topic which leads to state-of-the-art results in image processing applications, such as image denoising, inpainting or demosaicking [3,9,14]. Inspired by these studies, the aim of this paper is to explore how image sparse representations can be useful to efficiently calculate morphological operators.

Motivation and outline of the approach. In many practical situations, a collection of M binary or grey-level images (each image having N pixels) should be analysed by applying the same morphological operator (or a series of operators) to each image. If one considers that the content of the various images is relatively similar, we can expect that the initial collection can be efficiently projected into a dimensionality reduced image space. Then, the morphological operator (or an equivalent operator) can be applied to the reduced set of images of the projective space, in such a way that the original processed image is approximately obtained by projecting back to the initial space. Typical examples of image families which can be fit in this framework are: i) collection of shapes or a database of face images, ii) the spectral bands of a hyperspectral image, iii) the set of patches of a large image. The rationale behind this kind of approach is the hypothesis that the intrinsic dimension of the image collection is lower than $N \times M$. Usually the subspace representation involves deriving a set of basis components (or dictionary composed of atoms) using linear techniques like PCA or ICA. The projection coefficients for the linear combinations in the above methods can be either positive or negative, and such linear combinations generally involve complex cancellations between positive and negative numbers. Therefore, these representations lack the intuitive meaning of "adding parts to form a whole". This property is particularly problematic in the case of mathematical morphology since the basic binary operator, the dilation of a set, is defined as the operator which commutes with the union of parts of the set. Non-negative matrix factorization (NMF) [6] imposes the non-negativity constraints in learning basis images: the pixels values of resulting images as well as the coefficients for the reconstruction are all non-negative. This ensures that NMF is a procedure for learning a parts-based representation [6]. In addition, sparse modelling involves for the construction of a succinct representation of some data as a combination of a few typical patterns (few atoms of the dictionary) learned from the data itself. Hence, the notion of sparse mathematical morphology introduced for the first time in this paper is based on sparse NMF.

Paper organisation. This paper is structured as follows. Section 2 reviews the notion of NMF and the various algorithms proposed in the state-of-the-art, including the sparse variants. The use of NMF representations for implementing sparse pseudo-morphological binary dilations/erosions is introduced in Section 3. Using the level set decomposition of numerical functions, the extension to grey-level images of this morphological sparse processing is tackled in Section 4. Conclusions and perspectives are finally given in Section 5.

2 Non-negative Matrix Factorization and Variants

Definition. Let us assume that our data consists of M vectors of N non-negative scalar variables. Denoting the column vector $\mathbf{v}_j$, $j = 1, \cdots, M$, the matrix of data is obtained as $\mathbf{V} = (\mathbf{v}_1, \cdots, \mathbf{v}_M)$ (each $\mathbf{v}_j$ is the j-th column of $\mathbf{V}$), with $|\mathbf{v}_j| = N$. If we analyze M images of N pixels, these images can be stored in linearized form, so that each image will be a column vector of the matrix.

Given the non-negative matrix $\mathbf{V} \in \mathbb{R}^{N \times M}$, $\mathbf{V}_{i,j} \geq 0$, NMF is a linear non-negative approximate data decomposition into the two matrices $\mathbf{W} \in \mathbb{R}^{N \times R}$ and $\mathbf{H} \in \mathbb{R}^{R \times M}$ such that

$$\mathbf{V} \approx \mathbf{W}\mathbf{H}, \ \text{ s.t. } \mathbf{W}_{i,k}, \mathbf{H}_{k,j} \geq 0, \tag{1}$$

where usually $R \ll M$ (dimensionality reduction). Each of the R columns of $\mathbf{W}$ contains a basis vector $\mathbf{w}_k$ and each row of $\mathbf{H}$ contains the coefficient vector (weights) $\mathbf{h}_j$ corresponding to vector $\mathbf{v}_j$: $\mathbf{v}_j = \sum_{k=1}^{R} \mathbf{w}_k \mathbf{H}_{k,j} = \mathbf{W}\mathbf{h}_j$. Using the modern terminology, the matrix $\mathbf{W}$ contains the dictionary and $\mathbf{H}$ the encoding.

A theoretical study of the properties of NMF representation has been achieved in [2] using geometric notions: NMF is interpreted as the problem of finding a simplicial cone which contains the data points in the positive orthant, or in other words, NMF is a conical coordinate transformation.

Algorithms for computing NMF. The factorization $\mathbf{V} \approx \mathbf{W}\mathbf{H}$ is not necessarily unique, and the optimal choice of matrices $\mathbf{W}$ and $\mathbf{H}$ depends on the cost function that minimizes the reconstruction error. The most widely used is the Euclidean distance: minimize $\|\mathbf{V} - \mathbf{W}\mathbf{H}\|_2^2 = \sum_{i,j} \left(\mathbf{V}_{i,j} - (\mathbf{W}\mathbf{H})_{i,j}\right)^2$ with respect to $\mathbf{W}$ and $\mathbf{H}$, subject to the constraints $\mathbf{W}, \mathbf{H} > 0$. Although the minimization problem is convex in $\mathbf{W}$ and $\mathbf{H}$ separately, it is not convex in both simultaneously. In [7] is proposed a multiplicative good performance algorithm to implement this optimization problem. They proved that the cost function is nonincreasing at the iteration and the algorithm converges at least to a local optimal solution. More precisely, the update rules for both matrices are:

$$\mathbf{H}_{k,j} \leftarrow \mathbf{H}_{k,j} \frac{(\mathbf{W}^T\mathbf{V})_{k,j}}{(\mathbf{W}^T\mathbf{W}\mathbf{H})_{k,j}}; \quad \mathbf{W}_{i,k} \leftarrow \mathbf{W}_{i,k} \frac{(\mathbf{V}\mathbf{H}^T)_{i,k}}{(\mathbf{W}\mathbf{H}\mathbf{H}^T)_{k,j}}$$

Another useful cost function, also considered in [7], is the Kullback-Leibler (KL) divergence, which leads also quite simple multiplicative update rules. In [8], it was proposed a variant of KL divergence NMF, which is named Local NMF (LNMF), aiming at learning spatially localized components (by minimizing the number of basis R to represent $\mathbf{V}$ and by maximizing the energy of each retained components) as well as imposing that different bases should be as orthogonal as possible (in order to minimize redundancy between the different bases). The multiplicative update rules for LNMF are given by

$$\mathbf{H}_{k,j} \leftarrow \sqrt{\mathbf{H}_{k,j} \sum_i \mathbf{V}_{i,j} \frac{\mathbf{W}_{i,k}}{(\mathbf{W}\mathbf{H})_{i,k}}}; \quad \mathbf{W}_{i,k} \leftarrow \mathbf{W}_{i,k} \frac{\sum_j \mathbf{V}_{i,j} \frac{\mathbf{H}_{k,j}}{(\mathbf{W}\mathbf{H})_{i,j}}}{\sum_j H_{k,j}}; \quad \mathbf{W}_{i,k} \leftarrow \frac{\mathbf{W}_{i,k}}{\sum_i \mathbf{W}_{i,k}}$$

NMF with sparseness constraints. A very powerful framework to add a degree of sparseness in the basis vectors $\mathbf{W}$ and/or the coefficients $\mathbf{H}$ was introduced in [5]. The sparseness measure σ of a vector $\mathbf{v} \in \mathbb{R}^{N \times 1}$ used in [5] is based on the relationship between the L_1 norm and the L_2 norm: $\sigma(\mathbf{v}) = \frac{\sqrt{N} - \|\mathbf{v}\|_1 / \|\mathbf{v}\|_2}{\sqrt{N} - 1}$. This function is maximal at 1 iff $\mathbf{v}$ contains only a single non-zero component, and takes a value of 0 iff all components are equal (up to signs). Then, matrix $\mathbf{W}$ and $\mathbf{H}$ are solved by the problem (1) under additional constraints $\sigma(\mathbf{w}_k) = S_w$ and $\sigma(\mathbf{h}_j) = S_h$, where S_w and S_h are respectively the desired sparseness of $\mathbf{W}$ and $\mathbf{H}$. The algorithm introduced in [5] is a projected gradient descent algorithm (additive update rule), which takes a step in the direction of the negative gradient, and subsequently projects onto the constraint space. The most sophisticated step finds, for a given vector $\mathbf{v}$, the closest non-negative vector $\mathbf{u}$ with a given L_1 norm and a given L_2 norm, see technical details in [5]. Sparseness is controlled explicitly with a pair of parameters that is easily interpreted; in addition, the number of required iterations grows very slowly with the dimensionality of the problem. In fact, for all the empirical tests considered in this paper, we have used the MATLAB code for performing NMF and its various extensions (LNMF, sparse NMF) provided by P. Hoyer [5].

Besides the spareness parameters (S_w, S_h), a crucial parameter to be chosen in any NMF algorithm is the value of R, that is, the number of basis of projective reduced space. Any dimensionality reduction technique, such as PCA, requires also to fix the number of components. In PCA, the components are ranked according to the second-order statistical importance of the components and each one has associated a value of the represented variance; whereas in NMF the selection R can be evaluated only *a posteriori*, by evaluating the error of reconstruction.

3 Sparse Binary Pseudo-dilations and Pseudo-erosions

Let $\mathcal{X} = \{X_1, \cdots, X_M\}$ be a collection of M binary shapes, i.e., $X_j \in \mathcal{P}(E)$. For each shape X_j, let $\mathbf{x}_j(i) : I \to \{0,1\}$, with $i \in I = \{1, 2, \cdots, N\}$ and $N = |E|$, be its *characteristic vector*: $\forall X_j \in \mathcal{P}(E)$, we have $\mathbf{x}_j(i) = 1$ if $p_i \in X_j$ and $\mathbf{x}_j(i) = 0$ if $p_i \in X_j^c$. Then the shape family $\mathcal{X}$ has associated a data matrix $\mathbf{V} \in \{0,1\}^{N \times M}$, where each characteristic vector corresponds to one column, i.e., $\mathbf{V}_{i,j} = \mathbf{x}_j(i)$.

Sparse NMF approximations to binary sets. After computing NMF optimization on data $\mathbf{V}$, for a given dimensionality R, an approximation to $\mathbf{V}$ is obtained. More precisely, if we denote by $\phi_k(p_i) : E \to \mathbb{R}^+$ the basis images associated to the basis matrix $\mathbf{W}$, i.e., $\phi_k(p_i) = \mathbf{W}_{i,k}$, the following image is obtained as

$$a_{X_j}(p_i) = \sum_{k=1}^{R} \phi_k(p_i) \mathbf{H}_{k,j} \tag{2}$$

It is obvious that without any additional constrains, function $a_X(p_i)$ is neither a binary image not even strictly defined in interval $[0,1]$ (the value can be slightly

> 1). Hence, a thresholding operation at value α is required to impose a binary approximate set $\widetilde{X}_j$ to each initial shape X_j, i.e.,

$$X_j \xrightarrow{NMF} \widetilde{X}_j : \quad p_i \in \widetilde{X}_j \quad \text{if} \quad a_X(p_i) > \alpha \tag{3}$$

We propose to fix, for all the examples of the paper, the threshold value to $\alpha = 0.4$, in order to favor the reconstruction of X_j against its complement.

Let us consider a practical example of binary image collection $\mathcal{X}$, using $M = 100$ images of the Fish shape database ($N = 400 \times 200$). Fig. 1 depicts the corresponding basis images for various NMF algorithms: we have fixed $R = 10$ for all the cases (relatively strong dimensionality reduction). We observe that standard NMF produces a partial part-based representation, which includes also almost complete objects for outlier shapes (basis 2-upper-center and 5-center-center). As expected, LNMF produces more local decompositions, however the orthogonality constraints involves also an atomization of some common parts. A similar problem arises for Sparse-NMF when $S_w \neq 0$ (constraint of sparsity in basis matrix $\mathbf{W}$). When the sparsity constraint is limited to S_h, with a typical value around 0.6, the obtained dictionary of shapes is less local, but in exchange, this constraint involves that each binary shapes is reconstructed using a limited number of atoms. The various groups of fish shapes are therefore better approximated by the latter case than using the other NMF algorithms. The comparison of Fig. 2 illustrates the better performance of Sparse-NMF ($S_w = 0$, $S_h = 0.6$) with respect to the others.

Sparse max-approximation to binary dilation. The two fundamental morphological operators are the dilation and the erosion, which are defined respectively as the operators which preserve the union and the intersection. Given a *structuring element* $B \subseteq E$, i.e., a set defined at the origin which introduces the shape/size of the operator, the *dilation of a binary image* X by B and the *erosion of a binary image* X by B are defined respectively by by [11,4,1]

$$\delta_B(X) = \cup \{B(p_i) |\, p_i \in X\} , \tag{4}$$

and

$$\varepsilon_B(X) = \{p_i \in E |\, B(p_i) \subseteq X\} , \tag{5}$$

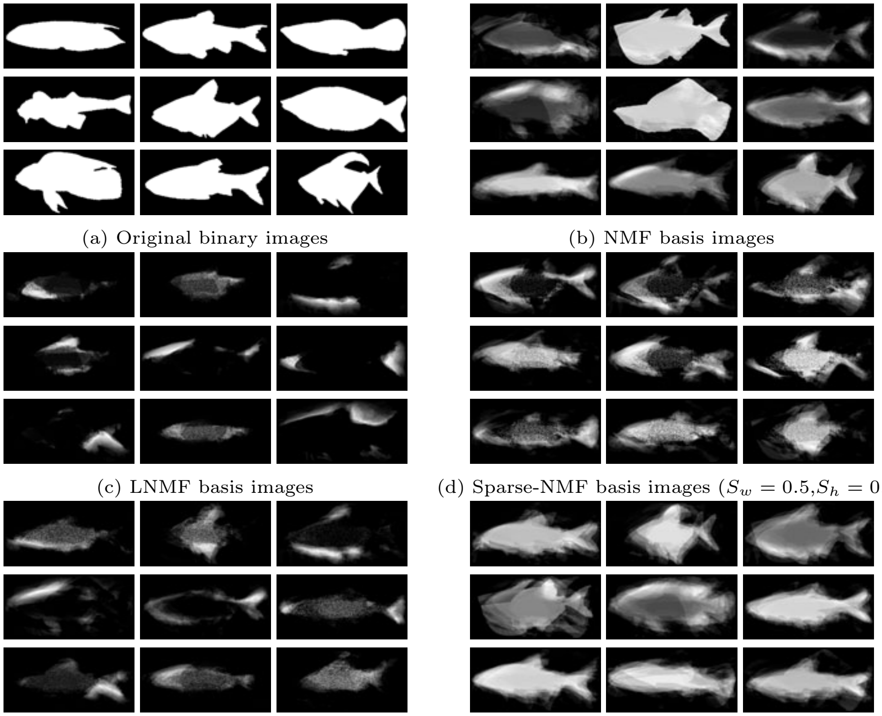
where $B(p_i)$ is the structuring element centered at pixel p_i. In the case of numerical functions $f \in \mathcal{F}(E, \mathcal{T})$, which are considered in detail in next section, the *dilation of a grey-level image* is defined by [4,12,1]

$$\delta_B(f)(p_i) = \left\{ f(p_m) |\, f(p_m) = \sup\left[f(p_n)\right], p_n \in \check{B}(p_i) \right\} , \tag{6}$$

and the dual *grey-level erosion* is given by [4,12,1]

$$\varepsilon_B(f)(p_i) = \{ f(p_m) |\, f(p_m) = \inf\left[f(p_n)\right], p_n \in B(p_i) \} . \tag{7}$$

where $\check{B}(p_i)$ is the transposed structuring element centered at pixel p_i. If B is symmetric with respect to the origin $\check{B} = B$.



(a) Original binary images

(b) NMF basis images

(c) LNMF basis images

(d) Sparse-NMF basis images ($S_w = 0.5$,$S_h = 0.5$)

(e) Sparse-NMF basis images ($S_w = 0.6$,$S_h = 0$)

(f) Sparse-NMF basis images ($S_w = 0$,$S_h = 0.6$)

Fig. 1. Non-negative representation of binary shapes. A collection of $M = 100$ shapes has been used in the NMF experiments (in (a) is given only a selection of 9 shapes), where the number of reduced dimensions has been fixed to $R = 10$ (in the examples are given the first 9 basis images).

The *characteristic function* of set X, denoted $\xi_X : E \rightarrow \{0, 1\}$, is defined by

$$\forall X \in \mathcal{P}(E),\ \forall p_i \in E,\quad \xi_X(p_i) = \begin{cases} 1 \text{ if } p_i \in X \\ 0 \text{ if } p_i \in X^c \end{cases} \tag{8}$$

For a function $f : E \rightarrow \mathcal{T}$, the *thresholded set* at value $t \in \mathcal{T}$ is a mapping from $\mathcal{F}(E, \mathcal{T})$ to $\mathcal{P}(E)$ given by [11]

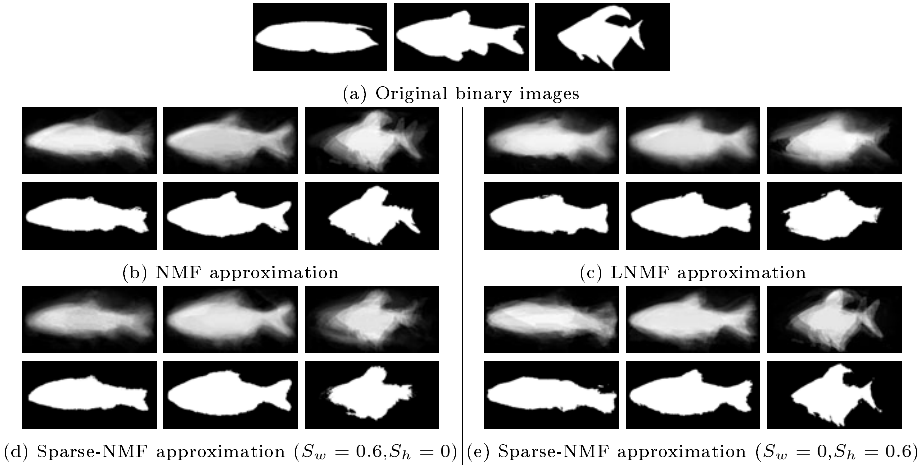
$$\varpi_t(f) = \{p_i \in E |\ f(p_i) \geq t\}\,. \tag{9}$$

Using these transformations it is obvious that the binary dilation (5) can be computed using the numerical operator (6), i.e.,

$$\delta_B(X) = \varpi_1\left(\delta_B(\xi_X)(p_i)\right). \tag{10}$$

We know that given a set defined as the union of a family of sets, i.e., $X = \cup_{k\in K} X_k$, the corresponding dilation is

$$\delta_B(X) = \delta_B\left(\cup_{k\in K} X_k\right) = \cup_{k\in K}\delta_B(X_k). \tag{11}$$



(a) Original binary images

(b) NMF approximation

(c) LNMF approximation

(d) Sparse-NMF approximation ($S_w = 0.6$,$S_h = 0$)

(e) Sparse-NMF approximation ($S_w = 0$,$S_h = 0.6$)

Fig. 2. Sparse NMF approximations to binary sets: (a) three original shapes X_j; (b)-(e) Top, reconstructed function a_{X_j} and Bottom, approximate set $\widetilde{X}_j$

It is also easy to see that

$$\delta_B(X) = \varpi_1\left(\delta_B\left(\min\left\{1, \sum_{k\in K}\xi_{X_k}(p_i)\right\}\right)\right) = \varpi_1\left(\sum_{k\in K}\delta_B\left(\xi_{X_k}(p_i)\right)\right). \quad (12)$$

In fact, the justification for using NMF in sparse mathematical morphology arises from equations (11) and (12).

Coming back to the NMF reconstruction, expressions (2) and (3), we can now write

$$X_j \approx \widetilde{X}_j = \varpi_\alpha\left(\sum_{k=1}^{R}\phi_k(p_i)\mathbf{H}_{k,j}\right). \quad (13)$$

Hence, we propose to introduce the following nonlinear operator, named *sparse max-approximation to binary dilation*,

$$D_B(X_j) = \varpi_\alpha\left(\sum_{k=1}^{R}\delta_B\left(\phi_k\right)(p_i)\mathbf{H}_{k,j}\right). \quad (14)$$

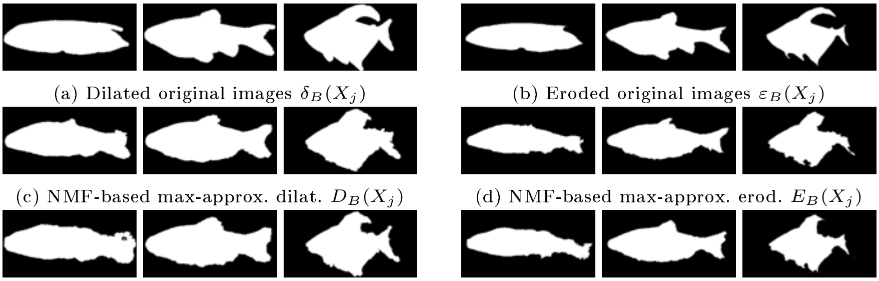
Note that by the positivity of $\mathbf{H}_{k,j}$, we have $\delta_B\left(\phi_k(p_i)\mathbf{H}_{k,j}\right) = \delta_B\left(\phi_k\right)(p_i)\mathbf{H}_{k,j}$. We can say that $\delta_B(X_j) \approx D_B(X_j)$, however neither the increasiness nor the extensitivity of $D_B(X_j)$ w.r.t. X_j can be guaranteed and consequently, this operator is not a morphological dilation. In other terms, in order to approximate the dilation by B of any of the M sets X_j, we only need to calculate the dilation of the R basis images. In addition, the sparsity of $\mathbf{H}$ involves that only a limited number of dilated atoms are required for each X_j.

Dual sparse max-approximation to binary erosion. One of the most interesting properties of mathematical morphology is the duality by the complement of pair of operators. Hence, the binary erosion of set X by B can be defined as the dual operator to the dilation: $\varepsilon_{\check{B}}(X) = (\delta_B(X^c))^c$. Using this property, we propose to define the *sparse max-approximation to binary erosion* as

$$E_B(X_j) = \varpi_\alpha \left(\sum_{k=1}^{R} \mathsf{C}\left[\delta_B \left(\mathsf{C}[\phi_k] \right)(p_i) \right] \mathbf{H}_{k,j} \right), \qquad (15)$$

where the complement basis images are defined by $\mathsf{C}[\phi_k(p_i)] = \max(\mathbf{W}_{i,k}) - \phi_k(p_i) + \min(\mathbf{W}_{i,k})$.

The results of $D_B(X_j)$ and $E_B(X_j)$ for three examples of the Fish shapes, compared with respect to the standard binary dilation and erosion, are given in Fig. 3. We have compared in particular the sparse max-approximation for the standard NMF and for the Sparse-NMF.



(a) Dilated original images $\delta_B(X_j)$ (b) Eroded original images $\varepsilon_B(X_j)$

(c) NMF-based max-approx. dilat. $D_B(X_j)$ (d) NMF-based max-approx. erod. $E_B(X_j)$

(e) Sparse NMF-based max-approx. dilat. $D_B(X_j)$ (f) Sparse NMF-based max-approx. erod. $E_B(X_j)$

Fig. 3. Comparison of dilation/erosion (a)/(b) vs. sparse pseudo operators for three examples of the Fish shapes. It is compared in particular the sparse max-approximation to dilation/erosion for the standard NMF (c)/(d) and for the Sparse-NMF (e)/(f), with ($S_w = 0$,$S_h = 0.6$). The structuring element B is a square of 5×5 pixels.

4 Extension to Grey-Level Images

We deal in this section with families of discrete grey-level images, i.e., $\mathcal{F} = \{f_1(p_i), \cdots, f_M(p_i)\}$, with $f_j(p_i) \in \mathcal{F}(E, \mathcal{T})$, $\mathcal{T} = \{t_1, t_2, \cdots, t_L\}$ with $(t_{l+1} - t_l) = \Delta t$. The thresholded set of f_j at each t_l, i.e., $X_j^{t_l} = \varpi_{t_l}(f_j)$, is called the cross-section or level-set at t_l. The set of cross-sections constitutes a family of decreasing sets: $t_\lambda \geq t_\mu \Rightarrow X^{t_\lambda} \subseteq X^{t_\mu}$ and $X^{t_\lambda} = \cap\{X^{t_\mu}, \mu < \lambda\}$. Any image f_j can be viewed as an unique stack of its cross-sections, which leads to the following reconstruction property:

$$f_j(p_i) = \sup\{t_l |\, p_i \in X_j^{t_l}\}, \quad t_l \in \mathcal{T}. \qquad (16)$$

We prefer here to consider the alternative reconstruction [13,10] using a numerical sum of the characteristic function of cross sections:

$$f_j(p_i) = \Delta t \sum_{l=1}^{L} \xi_{X_j^{t_l}}(p_i). \qquad (17)$$

It is well known in mathematical morphology that any binary increasing operator, such as the dilation and erosion, can be generalised to grey-level images by applying the binary operator to each cross-section, and then by reconstructing the corresponding grey-level image [11,10], i.e.,

$$\delta_B(f_j)(p_i) = \Delta t \sum_{l=1}^{L} \xi_{\delta_B\left(X_j^{t_l}\right)}(p_i), \quad \text{and} \quad \varepsilon_B(f_j)(p_i) = \Delta t \sum_{l=1}^{L} \xi_{\varepsilon_B\left(X_j^{t_l}\right)}(p_i). \quad (18)$$

Consider now that each image of the initial grey-level family $\mathcal{F}$ of M images is decomposed into its L cross-sections. Hence, we have

$$\mathcal{F} = \{f_1, \cdots, f_M\} \mapsto \mathcal{X} = \{X_1^{t_1}, X_1^{t_2} \cdots, X_1^{t_L}, \cdots, X_{M-1}^{t_L}, X_M^{t_1}, \cdots, X_M^{t_L}\},$$

where $\mathcal{X}$ is a family of $M' = M \times L$ binary images. Therefore, we can use NMF algorithms, for a given dimension R, to approximate each set $X_j^{t_l}$ and then approximate the corresponding function $f_j(p_i)$. Using the results of the previous section, we are able now to introduce the following definition for the *sparse max-approximation to grey-level dilation and erosion* given respectively by:

$$D_B(f_j)(p_i) = \Delta t \sum_{l=1}^{L} \xi_{D_B\left(X_j^{t_l}\right)}(p_i), \quad (19)$$

and

$$E_B(f_j)(p_i) = \Delta t \sum_{l=1}^{L} \xi_{E_B\left(X_j^{t_l}\right)}(p_i). \quad (20)$$

with

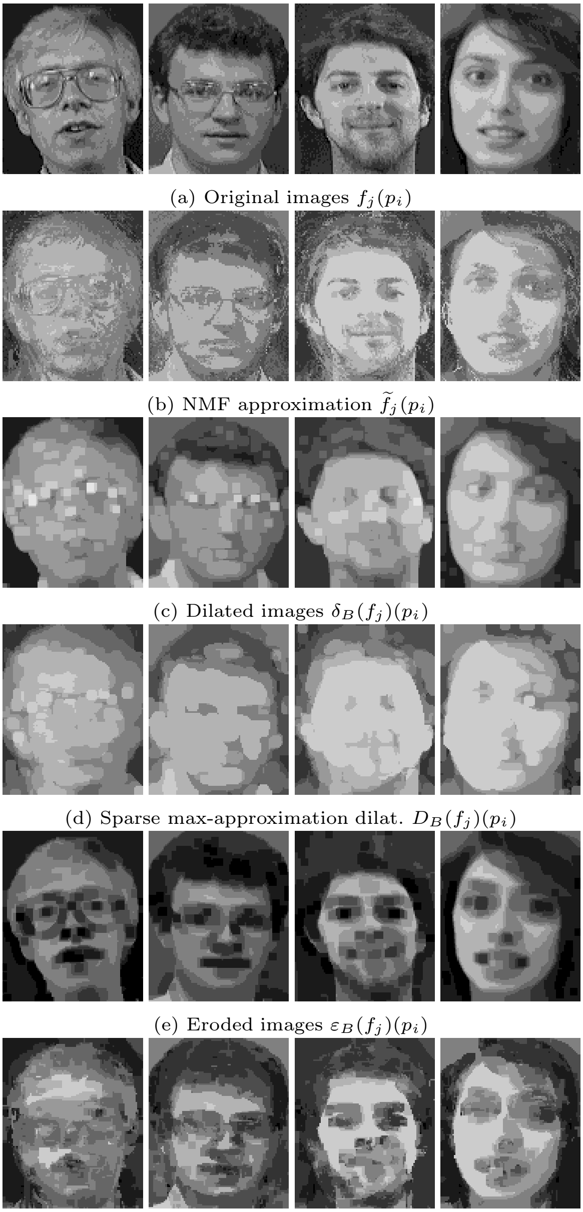
$$D_B\left(X_j^{t_l}\right) = \varpi_\alpha \left(\sum_{k=1}^{R} \delta\left(\phi_k\right)(p_i) \mathbf{H}_{k,j+l} \right);$$

$$E_B\left(X_j^{t_l}\right) = \varpi_\alpha \left(\sum_{k=1}^{R} \complement\left[\delta_B\left(\complement[\phi_k]\right)(p_i)\right] \mathbf{H}_{k,j+l} \right).$$

We must point out again that these approximate nonlinear operators do not satisfy the standard properties of grey-level dilation and erosion. The precise properties of $D_B(f_j)$ and $E_B(f_j)$ will be studied in detail in a subsequent paper.

Fig. 4 illustrates the behaviour of sparse max-approximation to grey-level dilation and erosion. This preliminary experiment deals with images from the ORL face database. In this particular case, the results obtained using the standard NMF are significantly better than those obtained using any Sparse-NMF version.

In fact, as we can observe, the quality of the sparse max-approximation to dilation and to erosion, for a particular image, depends on the quality of the initial NMF reconstruction of the image. For instance, the first face (man with glasses) is not very well approximated with the learned NMF basis and hence, its approximated dilation and erosion are also unsatisfactory. On the contrary, in the case of the last image (woman), the results are more appropriate.



(a) Original images $f_j(p_i)$

(b) NMF approximation $\widetilde{f}_j(p_i)$

(c) Dilated images $\delta_B(f_j)(p_i)$

(d) Sparse max-approximation dilat. $D_B(f_j)(p_i)$

(e) Eroded images $\varepsilon_B(f_j)(p_i)$

(f) Sparse max-approximation erod. $E_B(f_j)(p_i)$

Fig. 4. Four examples of the ORL face database (a) (quantized in $L = 10$ grey-levels) and corresponding approximated image using standard NMF. Comparison of dilation/erosion (c)/(d) vs. sparse max-approximation to dilation/erosion for the standard NMF (e)/(f). A collection of $M = 20$ faces has been used in the NMF experiments where the number of reduced dimensions for the binary matrix $\mathbf{V}$ has been fixed to $R = 75$ (note that $M' = 20 \times 10 = 200$ dimensions). The structuring element B is a square of size 3×3 pixels.

In any case, this is only a first experiment and a more concise investigation of the effect of variations of the parameters would be studied in ongoing research. Typically, we need to evaluate quantitatively, for a given operator, the approximation power for different sizes of structuring element as well as the approximation power of the morphological operators with respect to the degree of approximation of the initial image. In addition, the influence on the approximated morphological operators of dimensionality R and of the parameters S_w and S_h from Sparse NMF should be also evaluated. It will be also probably useful a direct comparison to another dimensionality reduction technique, typically PCA, in order to have a better judgment of the potential of the presented approach.

5 Conclusions and Perspectives

We have introduced the notion of sparse binary and grey-level pseudo-dilations and erosions using NMF sparse representation. The first results are relatively encouraging and they open a new avenue to study how the current paradigm of sparse modelling (based mainly on linear operations) can be particularised to the nonlinear morphological framework.

Besides a deeper experimental quantitative analysis, as discussed above, in ongoing research, we will focus, on the one hand, on NMF representations which produces binary basis images, see for instance [15], which is still a quite open problem. On the other hand, we will study in deep the properties of the sparse max-approximation to dilation/erosion, as well as consider alternative extensions to the grey-level case. Finally, the construction of more complex operators than dilation/erosion, and eventually the introduction of new ones, and their possible applications for inverse problems (regularisation, debluring, etc.) will be also foreseen.

References

1. Bloch, I., Heijmans, H., Ronse, C.: Mathematical Morphology. In: Aiello, M., Pratt-Hartmann, I., van Benthem, J. (eds.) Handbook of Spatial Logics, ch. 14, pp. 857–944. Springer, Heidelberg (2007)
2. Donoho, D., Stodden, V.: When does non-negative matrix factorization give a correct decomposition into parts? In: Advances in Neural Information Processing 16 (Proc. NIPS 2003). MIT Press, Cambridge (2004)
3. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. on Image Proc. 15(12), 3736–3745 (2006)
4. Heijmans, H.J.A.M.: Morphological Image Operators. Academic Press, Boston (1994)
5. Hoyer, P.: Non-negative Matrix Factorization with Sparseness Constraints. J. Mach. Learn. Res. 5, 1457–1469 (2004)
6. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (1999)

7. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing 13 (Proc. NIPS 2000). MIT Press, Cambridge (2001)
8. Li, S.Z., Hou, X., Zhang, H., Cheng, Q.: Learning spatially localized parts-based representations. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Hawaii, USA, vol. I, pp. 207–212 (2001)
9. Mairal, J., Elad, M., Sapiro, G.: Sparse representation for color image restoration. IEEE Trans. on Image Proc. 17(1), 53–69 (2008)
10. Ronse, C.: Bounded variation in posets, with applications in morphological image processing. In: Passare, M. (ed.) Proceedings of the Kiselmanfest 2006, Acta Universitatis Upsaliensis, vol. 86, pp. 249–281 (2009)
11. Serra, J.: Image Analysis and Mathematical Morphology. Image Analysis and Mathematical Morphology, vol. I. Theoretical Advances, vol. II. Academic Press, London (1982) (1988)
12. Soille, P.: Morphological Image Analysis. Springer, Berlin (1999)
13. Wendt, P.D., Coyle, E.J., Gallagher, N.C.: Stack Filters. IEEE Trans. on Acoustics, Speech, and Signal Processing 34(4), 898–911 (1986)
14. Yu, G., Sapiro, G., Mallat, S.: Solving Inverse Problems with Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity. IEEE Trans. on Image Processing (2011)
15. Yuan, Y., Li, X., Pang, Y., Lu, X., Tao, D.: Binary Sparse Nonnegative Matrix Factorization. IEEE Trans. on Circuits and Systems for Video Technology 19(5), 772–777 (2009)

Fuzzy Bipolar Mathematical Morphology: A General Algebraic Setting

Isabelle Bloch

Télécom ParisTech - CNRS LTCI - Paris, France
isabelle.bloch@telecom-paristech.fr

Abstract. Bipolar information is an important component in information processing, to handle both positive information (e.g. preferences) and negative information (e.g. constraints) in an asymmetric way. In this paper, a general algebraic framework is proposed to handle such information using mathematical morphology operators, leading to results that apply to any partial ordering.

Keywords: Bipolar information, fuzzy bipolar dilation and erosion, bipolar connectives.

1 Introduction

A recent trend in contemporary information processing focuses on bipolar information, both from a knowledge representation point of view, and from a processing and reasoning one. Bipolarity is important to distinguish between (i) positive information, which represents what is guaranteed to be possible, for instance because it has already been observed or experienced, and (ii) negative information, which represents what is impossible or forbidden, or surely false [1]. In this paper, we propose to handle such bipolar information using mathematical morphology operators. Mathematical morphology on bipolar fuzzy sets was proposed for the first time in [2], by considering the complete lattice defined from the Pareto ordering. Then it was further developed, with additional properties, geometric aspects and applications to spatial reasoning, in [3,4]. The lexicographic ordering was considered too in [5]. Here we propose a more general algebraic setting and we show that the usual properties considered in mathematical morphology hold in any complete lattice representing bipolar information, whatever the choice of the partial ordering. Recently, mathematical morphology on interval-valued fuzzy sets and intuitionistic fuzzy sets was addressed, independently, in [6], but without considering the algebraic framework of adjunctions, thus leading to weaker properties. This group then extended its approach with more properties in [7]. Pareto ordering was used in this work. Again this paper proposes a more general and powerful setting.

Mathematical morphology [8] usually relies on the algebraic framework of complete lattices [9]. Although it has also been extended to complete semilattices and general posets [10], based on the notion of adjunction [11], in this

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 13–24, 2011.

paper we only consider the case of complete lattices. Let us assume that bipolar information is represented by a pair (μ, ν), where μ represents the positive information and ν the negative information, under a consistency constraint [1]. Let us denote by $\mathcal{B}$ the set of all (μ, ν). We assume that it is possible to define a spatial ordering $\preceq$ on $\mathcal{B}$ such that $(\mathcal{B}, \preceq)$ is a complete lattice. We denote by $\bigvee$ and $\bigwedge$ the supremum and infimum, respectively. Once we have a complete lattice, it is easy to define algebraic dilations δ and erosions ε on this lattice, as classically done in mathematical morphology [11,12], as operations that commute with the supremum and with the infimum, respectively:

$$\forall (\mu_i, \nu_i) \in \mathcal{B},\ \delta(\bigvee_i (\mu_i, \nu_i)) = \bigvee_i \delta((\mu_i, \nu_i)), \tag{1}$$

$$\forall (\mu_i, \nu_i) \in \mathcal{B},\ \varepsilon(\bigwedge_i (\mu_i, \nu_i)) = \bigwedge_i \varepsilon((\mu_i, \nu_i)), \tag{2}$$

where (μ_i, ν_i) is any family (finite or not) of elements of $\mathcal{B}$.

Classical results derived from the properties of complete lattices and adjunctions [11,12] hold in the bipolar case too.

Bipolar information can be represented in different frameworks, leading to different forms of μ and ν, for instance:

- positive and negative information are subsets P and N of some set, and the consistency constraint is expressed as $P \cap N = \emptyset$, expressing that what is possible or preferred (positive information) should be included in what is not forbidden (negative information) [1];
- μ and ν are membership functions to fuzzy sets, defined over a space $\mathcal{S}$, and the consistency constraint is expressed as $\forall x \in \mathcal{S}, \mu(x) + \nu(x) \leq 1$ [2]. The pair (μ, ν) is then called a bipolar fuzzy set;
- positive and negative information are represented by logical formulas φ and ψ, generated by a set of propositional symbols and connectives, and the consistency constraint is then expressed as $\varphi \wedge \psi \models \perp$ (ψ represents what is forbidden or impossible).
- Other examples include functions such as utility functions or capacities [13], preference functions [14], four-valued logics [15], possibility distributions [16].

One of the main issues in the proposed extensions of mathematical morphology to bipolar information is to handle the two components (i.e. positive and negative information) and to define an adequate and relevant ordering. Two extreme cases are Pareto ordering (also called marginal ordering) and lexicographic ordering. The Pareto ordering handles both components in a symmetric way, while the lexicographic ordering on the contrary gives a strong priority to one component, and the other one is then seldom considered. This issue has been addressed in other types of work, where different partial orderings have been discussed, such as color image processing (see e.g. [17]) and social choice (see e.g. [18]). The works in these domains, and the various partial orderings proposed, can guide the choice of an ordering adapted to bipolar information.

In the following, we will detail the case of bipolar fuzzy sets, extending our previous work in [2,3] to any partial ordering. This includes the other examples described above: the case of sets corresponds to the case where only bipolarity should be taken into account, without fuzziness (hence the membership functions take only values 0 and 1). In the case of logical formulas, we consider the models $[\![\varphi]\!]$ and $[\![\psi]\!]$ as sets or fuzzy sets. Hence the case of bipolar fuzzy sets is general enough to cover several other mathematical settings.

The lattice structure is described in Section 2. Then bipolar connectives and their properties are detailed in Section 3. They are then used to define general forms of morphological dilations and erosions in Section 4, based on bipolar degrees of intersection and inclusion. Proofs are omitted here, and can be found in [19].

2 Partial Ordering and Lattice of Bipolar Fuzzy Sets

Let $\mathcal{S}$ be the underlying space (the spatial domain for spatial information processing). A bipolar fuzzy set on $\mathcal{S}$ is defined by an ordered pair of functions (μ, ν) such that $\forall x \in \mathcal{S}, \mu(x) + \nu(x) \leq 1$. Note that bipolar fuzzy sets are formally linked to intuitionistic fuzzy sets [20], interval-valued fuzzy sets [21] and vague sets, where the interval at each point x is $[\mu(x), 1-\nu(x)]$, or to clouds when boundary constraints are added [22], as shown by several authors [23]. However their respective semantics are very different, and we keep here the terminology of bipolarity, for handling asymmetric bipolar information [16].

For each point x, $\mu(x)$ defines the degree to which x belongs to the bipolar fuzzy set (positive information) and $\nu(x)$ the non-membership degree (negative information). This formalism allows representing both bipolarity and fuzziness. The set of bipolar fuzzy sets defined on $\mathcal{S}$ is denoted by $\mathcal{B}$.

Let us denote by $\mathcal{L}$ the set of ordered pairs of numbers (a, b) in $[0, 1]$ such that $a + b \leq 1$ (hence $(\mu, \nu) \in \mathcal{B} \Leftrightarrow \forall x \in \mathcal{S}, (\mu(x), \nu(x)) \in \mathcal{L}$). Let $\preceq$ be a partial ordering on $\mathcal{L}$ such that $(\mathcal{L}, \preceq)$ is a complete lattice. We denote by $\bigvee$ and $\bigwedge$ the supremum and infimum, respectively. The smallest element is denoted by $0_\mathcal{L}$ and the largest element by $1_\mathcal{L}$.

The partial ordering on $\mathcal{L}$ induces a partial ordering on $\mathcal{B}$, also denoted by $\preceq$ for the sake of simplicity:

$$(\mu_1, \nu_1) \preceq (\mu_2, \nu_2) \text{ iff } \forall x \in \mathcal{S}, (\mu_1(x), \nu_1(x)) \preceq (\mu_2(x), \nu_2(x)). \tag{3}$$

Then $(\mathcal{B}, \preceq)$ is a complete lattice, for which the supremum and infimum are also denoted by $\bigvee$ and $\bigwedge$. The smallest element is the bipolar fuzzy set (μ_0, ν_0) taking value $0_\mathcal{L}$ at each point, and the largest element is the bipolar fuzzy set $(\mu_\mathbb{I}, \nu_\mathbb{I})$ always equal to $1_\mathcal{L}$. Note that the supremum and the infimum do not necessarily provide one of the input bipolar numbers or bipolar fuzzy sets (in particular if they are not comparable according to $\preceq$). However, they do in case $\preceq$ is a total ordering.

3 Bipolar Connectives

Let us now introduce some connectives, that will be useful in the following and that extend to the bipolar case the connectives classically used in fuzzy set theory. In all what follows, increasingness or decreasingness is intended according to the partial ordering $\preceq$.

A **bipolar negation, or complementation**, on $\mathcal{L}$ is a decreasing operator N such that $N(0_{\mathcal{L}}) = 1_{\mathcal{L}}$ and $N(1_{\mathcal{L}}) = 0_{\mathcal{L}}$. In this paper, we restrict ourselves to involutive negations, such that $\forall (a,b) \in \mathcal{L}, N(N((a,b))) = (a,b)$ (these are the most interesting ones for mathematical morphology).

A **bipolar conjunction** is an operator C from $\mathcal{L} \times \mathcal{L}$ into $\mathcal{L}$ such that $C(0_{\mathcal{L}}, 0_{\mathcal{L}}) = C(0_{\mathcal{L}}, 1_{\mathcal{L}}) = C(1_{\mathcal{L}}, 0_{\mathcal{L}}) = 0_{\mathcal{L}}$, $C(1_{\mathcal{L}}, 1_{\mathcal{L}}) = 1_{\mathcal{L}}$, and that is increasing in both arguments, i.e.: $\forall ((a_1,b_1),(a_2,b_2),(a'_1,b'_1),(a'_2,b'_2)) \in \mathcal{L}^4, (a_1,b_1) \preceq (a'_1,b'_1)$ and $(a_2,b_2) \preceq (a'_2,b'_2) \Rightarrow C((a_1,b_1),(a_2,b_2)) \preceq C((a'_1,b'_1),(a'_2,b'_2))$.

A **bipolar t-norm** is a commutative and associative bipolar conjunction such that $\forall (a,b) \in \mathcal{L}, C((a,b), 1_{\mathcal{L}}) = C(1_{\mathcal{L}}, (a,b)) = (a,b)$ (i.e. the largest element of $\mathcal{L}$ is the unit element of C). If only the property on the unit element holds, then C is called a **bipolar semi-norm**.

A **bipolar disjunction** is an operator D from $\mathcal{L} \times \mathcal{L}$ into $\mathcal{L}$ such that $D(1_{\mathcal{L}}, 1_{\mathcal{L}}) = D(0_{\mathcal{L}}, 1_{\mathcal{L}}) = D(1_{\mathcal{L}}, 0_{\mathcal{L}}) = 1_{\mathcal{L}}$, $D(0_{\mathcal{L}}, 0_{\mathcal{L}}) = 0_{\mathcal{L}}$, and that is increasing in both arguments.

A **bipolar t-conorm** is a commutative and associative bipolar disjunction such that $\forall (a,b) \in \mathcal{L}, D((a,b), 0_{\mathcal{L}}) = D(0_{\mathcal{L}}, (a,b)) = (a,b)$ (i.e. the smallest element of $\mathcal{L}$ is the unit element of D).

A **bipolar implication** is an operator I from $\mathcal{L} \times \mathcal{L}$ into $\mathcal{L}$ such that $I(0_{\mathcal{L}}, 0_{\mathcal{L}}) = I(0_{\mathcal{L}}, 1_{\mathcal{L}}) = I(1_{\mathcal{L}}, 1_{\mathcal{L}}) = 1_{\mathcal{L}}$, $I(1_{\mathcal{L}}, 0_{\mathcal{L}}) = 0_{\mathcal{L}}$ and that is decreasing in the first argument and increasing in the second argument.

Proposition 1. *Any bipolar conjunction C has a null element, which is the smallest element of $\mathcal{L}$: $\forall (a,b) \in \mathcal{L}, C((a,b), 0_{\mathcal{L}}) = C(0_{\mathcal{L}}, (a,b)) = 0_{\mathcal{L}}$. Similarly, any bipolar disjunction has a null element, which is the largest element of $\mathcal{L}$: $\forall (a,b) \in \mathcal{L}, D((a,b), 1_{\mathcal{L}}) = D(1_{\mathcal{L}}, (a,b)) = 1_{\mathcal{L}}$. For implications, we have $\forall (a,b) \in \mathcal{L}, I(0_{\mathcal{L}}, (a,b)) = I((a,b), 1_{\mathcal{L}}) = 1_{\mathcal{L}}$.*

As in the fuzzy case, conjunctions and implications may be related to each other based on the residuation principle, which corresponds to a notion of adjunction. This principle is expressed as follows in the bipolar case: a pair of bipolar connectives (I, C) forms an adjunction if, $\forall (a_i, b_i) \in \mathcal{L}, i = 1...3$,

$$C((a_1,b_1),(a_3,b_3)) \preceq (a_2,b_2) \Leftrightarrow (a_3,b_3) \preceq I((a_1,b_1),(a_2,b_2)). \tag{4}$$

These connectives can be linked to each other in different ways (again this is similar to the fuzzy case).

Proposition 2. *The following properties hold:*

- *Given a bipolar t-norm C and a bipolar negation N, the following operator D defines a bipolar t-conorm: $\forall ((a_1,b_1),(a_2,b_2)) \in \mathcal{L}^2$,*

$$D((a_1,b_1),(a_2,b_2)) = N(C(N((a_1,b_1)), N((a_2,b_2)))). \tag{5}$$

- *A bipolar implication I induces a bipolar negation N defined as:*

$$\forall(a,b) \in \mathcal{L}, N((a,b)) = I((a,b), 0_{\mathcal{L}}). \tag{6}$$

- *The following operator I_N, derived from a bipolar negation N and a bipolar conjunction C, defines a bipolar implication:* $\forall((a_1,b_1),(a_2,b_2)) \in \mathcal{L}^2$,

$$I_N((a_1,b_1),(a_2,b_2)) = N(C((a_1,b_1), N((a_2,b_2)))). \tag{7}$$

- *Conversely, a bipolar conjunction C can be defined from a bipolar negation N and a bipolar implication I:* $\forall((a_1,b_1),(a_2,b_2)) \in \mathcal{L}^2$,

$$C((a_1,b_1),(a_2,b_2)) = N(I((a_1,b_1), N((a_2,b_2)))). \tag{8}$$

- *Similarly, a bipolar implication can be defined from a negation N and a bipolar disjunction D as:* $\forall((a_1,b_1),(a_2,b_2)) \in \mathcal{L}^2$,

$$I_N((a_1,b_1),(a_2,b_2)) = D(N((a_1,b_1)),(a_2,b_2)). \tag{9}$$

- *A bipolar implication can also be defined by residuation from a bipolar conjunction C such that* $\forall(a,b) \in \mathcal{L} \backslash 0_{\mathcal{L}}, C(1_{\mathcal{L}},(a,b)) \neq 0_{\mathcal{L}}$*:* $\forall((a_1,b_1),(a_2,b_2)) \in \mathcal{L}^2$,

$$I_R((a_1,b_1),(a_2,b_2)) = \bigvee\{(a_3,b_3) \in \mathcal{L} \mid C((a_1,b_1),(a_3,b_3)) \preceq (a_2,b_2)\}.$$

  The operators C and I_R are then said to be adjoint (see the definition in Equation 4).
- *Conversely, from a bipolar implication I_R such that* $\forall(a,b) \in \mathcal{L} \setminus 1_{\mathcal{L}}, I_R(1_{\mathcal{L}},(a,b)) \neq 1_{\mathcal{L}}$*, the conjunction C such that (C, I_R) forms an adjunction is given by:* $\forall((a_1,b_1),(a_2,b_2)) \in \mathcal{L}^2$,

$$C((a_1,b_1),(a_2,b_2)) = \bigwedge\{(a_3,b_3) \in \mathcal{L} \mid (a_2,b_2) \preceq I_R((a_1,b_1),(a_3,b_3))\}.$$

Proposition 3. *If C and I are bipolar connectives such that (I,C) forms an adjunction (i.e. verifies Equation 4), then C distributes over the supremum and I over the infimum on the right, i.e.:* $\forall(a_i,b_i) \in \mathcal{L}, \forall(a,b) \in \mathcal{L}$,

$$\bigvee_i C((a,b),(a_i,b_i)) = C((a,b), \bigvee_i (a_i,b_i)), \quad \bigwedge_i I((a,b),(a_i,b_i)) = I((a,b), \bigwedge_i (a_i,b_i)).$$

Note that the distributivity on the left requires C to be commutative.

The following properties of adjunctions will also be useful for deriving mathematical morphology operators.

Proposition 4. *Let (I,C) be an adjunction. Then the following properties hold:*

- *C is increasing in the second argument and I in the second one. If furthermore C is commutative, then it is also increasing in the first one.*
- *$0_{\mathcal{L}}$ is the null element of C on the right and $1_{\mathcal{L}}$ is the null element of I on the right, i.e.* $\forall(a,b) \in \mathcal{L}, C((a,b), 0_{\mathcal{L}}) = 0_{\mathcal{L}},\ I((a,b), 1_{\mathcal{L}}) = 1_{\mathcal{L}}$.

4 Morphological Dilations and Erosions of Bipolar Fuzzy Sets

We now assume that $\mathcal{S}$ is an affine space (or at least a space on which translations can be defined), and we use the notion of structuring element, which defines a spatial neighborhood of each point in $\mathcal{S}$ (or a binary relation between worlds in a logical framework). Here we consider fuzzy bipolar structuring elements. More generally, without any assumption on the underlying domain $\mathcal{S}$, a structuring element is defined as a binary relation between two elements of $\mathcal{S}$ (i.e. y is in relation with x if and only if $y \in B_x$). This allows on the one hand dealing with spatially varying structuring elements (when $\mathcal{S}$ is the spatial domain), or with graph structures, and on the other hand establishing interesting links with several other domains, such as rough sets, formal logics, and, in the more general case where the morphological operations are defined from one set to another one, with Galois connections and formal concept analysis. The general principle underlying morphological erosions consists in translating the structuring element at every position in space and check if this translated structuring element is included in the original set [8]. This principle has also been used in the main extensions of mathematical morphology to fuzzy sets or to logics. Similarly, defining morphological erosions of bipolar fuzzy sets, using bipolar fuzzy structuring elements, requires to define a degree of inclusion between bipolar fuzzy sets. Such inclusion degrees have been proposed in the context of intuitionistic fuzzy sets [24]. With our notations, a degree of inclusion of a bipolar fuzzy set (μ', ν') in another bipolar fuzzy set (μ, ν) is defined as [2]:

$$\bigwedge_{x \in \mathcal{S}} I((\mu'(x), \nu'(x)), (\mu(x), \nu(x))) \tag{10}$$

where I is a bipolar implication, and a degree of intersection is defined as:

$$\bigvee_{x \in \mathcal{S}} C((\mu'(x), \nu'(x)), (\mu(x), \nu(x))) \tag{11}$$

where C is a bipolar conjunction. Note that both inclusion and intersection degrees are elements of $\mathcal{L}$, i.e. they are defined as bipolar degrees.

Based on these concepts, we can now propose a general definition for morphological erosions and dilations, thus extending our previous work in [2,3,5].

Definition 1. *Let (μ_B, ν_B) be a bipolar fuzzy structuring element (in $\mathcal{B}$). The erosion of any (μ, ν) in $\mathcal{B}$ by (μ_B, ν_B) is defined from a bipolar implication I as:*

$$\forall x \in \mathcal{S},\ \varepsilon_{(\mu_B,\nu_B)}((\mu, \nu))(x) = \bigwedge_{y \in \mathcal{S}} I((\mu_B(y-x), \nu_B(y-x)), (\mu(y), \nu(y))). \tag{12}$$

In this equation, $\mu_B(y-x)$ (respectively $\nu_B(y-x)$) represents the value at point y of the translation of μ_B (respectively ν_B) at point x.

Definition 2. *Let (μ_B, ν_B) be a bipolar fuzzy structuring element (in $\mathcal{B}$). The dilation of any (μ, ν) in $\mathcal{B}$ by (μ_B, ν_B) is defined from a bipolar conjunction C as:*

$$\delta_{(\mu_B,\nu_B)}((\mu,\nu))(x) = \bigvee_{y \in \mathcal{S}} C((\mu_B(x-y), \nu_B(x-y)), (\mu(y), \nu(y))). \tag{13}$$

Proposition 5. *Definitions 1 and 2 are consistent: they actually provide bipolar fuzzy sets of $\mathcal{B}$, i.e. $\forall(\mu,\nu) \in \mathcal{B}, \forall(\mu_B,\nu_B) \in \mathcal{B}, \delta_{(\mu_B,\nu_B)}((\mu,\nu)) \in \mathcal{B}$ and $\varepsilon_{(\mu_B,\nu_B)}((\mu,\nu)) \in \mathcal{B}$.*

Proposition 6. *In case the bipolar fuzzy sets are usual fuzzy sets (i.e. $\nu = 1-\mu$ and $\nu_B = 1 - \mu_B$), the definitions lead to the usual definitions of fuzzy dilations and erosions. Hence they are also compatible with classical morphology in case μ and μ_B are crisp.*

Proposition 7. *Definitions 1 and 2 provide an adjunction (ε, δ) if and only if (I, C) is an adjunction.*

Proposition 8. *If I and C are bipolar connectives such that (I, C) is an adjunction, then the operator ε defined from I by Equation 12 commutes with the infimum and the operator δ defined from C by Equation 13 commutes with the supremum, i.e. they are algebraic erosion and dilation. Moreover they are increasing with respect to (μ, ν).*

Proposition 9. *If (I, C) is an adjunction such that C is increasing in the first argument and I is decreasing in the first argument (typically if they are a bipolar conjunction and a bipolar implication), then the operator ε defined from I by Equation 12 is decreasing with respect to the bipolar fuzzy structuring element and the operator δ defined from C by Equation 13 is increasing with respect to the bipolar fuzzy structuring element.*

Proposition 10. *C distributes over the supremum and I over the infimum on the right if and only if ε and δ defined by Equations 12 and 13 are algebraic erosion and dilation, respectively.*

Proposition 11. *Let δ and ε be a dilation and an erosion defined by Equations 13 and 12. Then, for all $(\mu_B, \nu_B), (\mu, \nu), (\mu', \nu')$ in $\mathcal{B}$, we have:*

$$\delta_{(\mu_B,\nu_B)}((\mu,\nu) \wedge (\mu',\nu')) \preceq \delta_{(\mu_B,\nu_B)}((\mu,\nu)) \wedge \delta_{(\mu_B,\nu_B)}((\mu',\nu')), \tag{14}$$

$$\varepsilon_{(\mu_B,\nu_B)}((\mu,\nu)) \vee \varepsilon_{(\mu_B,\nu_B)}((\mu',\nu')) \preceq \varepsilon_{(\mu_B,\nu_B)}((\mu,\nu) \vee (\mu',\nu')). \tag{15}$$

Proposition 12. *A dilation δ defined by Equation 13 is increasing with respect to the bipolar fuzzy structuring element, while an erosion ε defined by Equation 12 is decreasing with respect to the bipolar fuzzy structuring element.*

These results fit well with the intuitive meaning behind the morphological operators. Indeed, a dilation is interpreted as a degree of intersection, which is easier to achieve with a larger structuring element, while an erosion is interpreted as a degree of inclusion, which means a stronger constraint if the structuring element is larger.

Proposition 13. *Let δ and ε be a dilation and an erosion defined by Equations 13 and 12. Then, for all $(\mu_B, \nu_B), (\mu'_B, \nu'_B), (\mu, \nu)$ in $\mathcal{B}$, we have:*

$$\delta_{(\mu_B,\nu_B)\wedge(\mu'_B,\nu'_B)}((\mu,\nu) \preceq \delta_{(\mu_B,\nu_B)}((\mu,\nu)) \wedge \delta_{(\mu'_B,\nu'_B)}((\mu,\nu)), \tag{16}$$

$$\varepsilon_{(\mu_B,\nu_B)}((\mu,\nu)) \vee \varepsilon_{(\mu'_B,\nu'_B)}((\mu,\nu)) \preceq \varepsilon_{(\mu_B,\nu_B)\wedge(\mu'_B,\nu'_B)}((\mu,\nu)). \tag{17}$$

Proposition 14. *Let δ be a dilation defined by Equation 13 from a bipolar conjunction C. The dilation satisfies $\delta_{(\mu_B,\nu_B)}((\mu,\nu)) = \delta_{(\mu,\nu)}((\mu_B,\nu_B))$ if and only if C is commutative.*

This result is quite intuitive. When interpreting the dilation as a degree of intersection, it is natural to expect this degree to be symmetrical in both arguments. Hence the commutativity of C has to be satisfied.

Proposition 15. *Let δ be a dilation defined by Equation 13 from a bipolar conjunction C. It satisfies the iterativity property, i.e.:*

$$\delta_{(\mu_B,\nu_B)}(\delta_{(\mu'_B,\nu'_B)}(\mu,\nu)) = \delta_{\delta_{(\mu_B,\nu_B)}((\mu'_B,\nu'_B))}((\mu,\nu)),$$

if and only if C is associative.

Proposition 16. *Let δ be a dilation defined by Equation 13 from a bipolar conjunction C. If C is a bipolar conjunction that admits $1_{\mathcal{L}}$ as unit element on the left (i.e. $\forall (a,b) \in \mathcal{L}, C(1_{\mathcal{L}}, (a,b)) = (a,b)$) and $C((a,b), 1_{\mathcal{L}}) \neq 1_{\mathcal{L}}$ for $(a,b) \neq 1_{\mathcal{L}}$, then the dilation is extensive, i.e. $\delta_{(\mu_B,\nu_B)}((\mu,\nu)) \succeq (\mu,\nu)$, if and only if $(\mu_B,\nu_B)(0) = 1_{\mathcal{L}}$, where 0 denotes the origin of space $\mathcal{S}$.*

A similar property holds for erosion and if I is a bipolar implication that admits $1_{\mathcal{L}}$ as unit element to the left (i.e. $\forall (a,b) \in \mathcal{L}, I(1_{\mathcal{L}}, (a,b)) = (a,b)$) and $I((a,b), 0_{\mathcal{L}}) \neq 0_{\mathcal{L}}$ for $(a,b) \neq 1_{\mathcal{L}}$, then the erosion is anti-extensive, i.e. $\varepsilon_{(\mu_B,\nu_B)}((\mu,\nu)) \preceq (\mu,\nu)$, if and only if $(\mu_B,\nu_B)(0) = 1_{\mathcal{L}}$.

The second condition on C holds in particular if $1_{\mathcal{L}}$ is also unit element on the right. This holds in specific cases in which C is a bipolar t-norm, which are the most interesting ones from a morphological point of view, as shown below.

Note that the condition $(\mu_B,\nu_B)(0) = 1_{\mathcal{L}}$ (i.e. the origin of space completely belongs to the bipolar fuzzy set, without any indetermination) is equivalent to the conditions on the structuring element found in classical [8] and fuzzy [25] morphology to have extensive dilations and anti-extensive erosions.

Proposition 17. *If I is derived from C and a negation N, then δ and ε are dual operators, i.e.: $\delta_{(\mu_B,\nu_B)}(N(\mu,\nu)) = N(\varepsilon_{(\check{\mu}_B,\check{\nu}_B)}((\mu,\nu)))$, where $(\check{\mu}_B,\check{\nu}_B)$ denotes the symmetrical of (μ_B,ν_B) with respect to the origin of $\mathcal{S}$.*

Duality with respect to complementation, which was advocated in the first developments of mathematical morphology [8], is important to handle in an consistent way an object and its complement for many applications (for instance in image processing and spatial reasoning). Therefore it is useful to know exactly under which conditions this property may hold, so as to choose the appropriate operators if it is needed for a specific problem. On the other hand, adjunction is a major feature of the "modern" view of mathematical morphology, with strong algebraic bases in the framework of complete lattices [9]. This framework is now widely considered as the most interesting one, since it provides consistent definitions with sound properties in different settings (continuous and discrete ones) and extending mathematical morphology to bipolar fuzzy sets in this framework inherits a set of powerful and important properties. Due to the interesting features of these two properties of duality and adjunction, in several applications both are required.

From all these results, we can derive the following theorem, which shows that the proposed forms are the most general ones for C being a bipolar t-norm.

Theorem 1. *Definition 2 defines a dilation with all properties of classical mathematical morphology if and only if C is a bipolar t-norm. The adjoint erosion is then defined by Equation 1 from the residual implication I_R derived from C. If the duality property is additionally required, then C and I have also to be dual operators with respect to a negation N.*

This important result shows that taking any conjunction may not lead to dilations that have nice properties. For instance the iterativity of dilation is of prime importance in concrete applications, and it requires associative conjunctions. This is actually a main contribution of our work, which differs from [6], where some morphological operators are suggested on intuitionistic fuzzy sets and for the Pareto ordering, but without referring to the algebraic framework, and leading to weaker properties (for instance the erosion defined in this work does not commute with the infimum and is then not an algebraic erosion). This group has then proposed some extensions in [7], still for the specific case of Pareto ordering, which closely follow our previous results in [2,3,5]. Moreover the result expressed in Theorem 1 is stronger and more general since it applies for any partial ordering leading to a complete lattice on $\mathcal{B}$. Note that pairs of adjoint operators are not necessarily dual. Therefore requiring both adjunction and duality properties may drastically reduce the choice for C and I. Note that this strong constraint is similar to the one proved for fuzzy sets in [26]. Although the choice of C and I is limited by the results expressed in Theorem 1 if sufficiently strong properties are required for the morphological operators, some choice may remain. The following property expresses a monotony property with respect to this choice.

Proposition 18. *Dilations and erosions are monotonous with respect to the choice of C and I:*

$$C \preceq C' \Rightarrow \delta^C \preceq \delta^{C'}$$

where δ^C is the dilation defined by Equation 13 using the bipolar conjunction or t-norm C, and

$$I \preceq I' \Rightarrow \varepsilon^I \preceq \varepsilon^{I'}$$

where ε^I is the erosion defined by Equation 12 using the bipolar implication I.

Examples of connectives and derived morphological operators, along with their properties, can be found for the Pareto ordering and for the lexicographic ordering in our previous work [2,3,5].

5 Conclusion

A general algebraic framework for fuzzy bipolar mathematical morphology was proposed, along with a set of properties. This general formulation is an original contribution, leading to new theoretical results. More properties on the compositions $\delta\varepsilon$ and $\varepsilon\delta$ can also be derived [19]. This framework can now be instantiated for various partial orderings. The case of Pareto ordering and lexicographic ordering have been detailed in [2,3,5], showing different properties, behaviors and interpretations.

From the basic morphological operators, other ones can be derived, as classically done in mathematical morphology, thus endowing the complete toolbox of operations with a bipolarity layer. Some examples of such operators (opening, closing, conditional operators, gradient...), along with geometrical measures and distances on bipolar fuzzy sets have been proposed in [4].

Let us now briefly comment on the applicability of these new tools for image processing and understanding. When dealing with spatial information, both fuzziness and bipolarity occur. Fuzziness may be related to the observed phenomenon itself, to the image acquisition process, to the image processing steps, to the knowledge used for image understanding and recognition, etc. This is now taken into account in a number of works. As for bipolarity, which has not been much addressed until now in this domain, several situations could benefit from its modeling. For instance, when assessing the position of an object in space, we may have positive information expressed as a set of possible places, and negative information expressed as a set of impossible or forbidden places (for instance because they are occupied by other objects). As another example, let us consider spatial relations. Human beings consider “left” and “right” as opposite relations. But this does not mean that one of them is the negation of the other one. The semantics of “opposite” captures a notion of symmetry (with respect to some axis or plane) rather than a strict complementation. In particular, there may be positions which are considered neither to the right nor to the left of some reference object, thus leaving room for some indetermination, neutrality or indifference.

As an illustrative example, a typical scenario showing the interest of bipolar representations of spatial relations and of morphological operations on these representations for spatial reasoning has been described in [3,4], for recognizing brain structures in medical images. The recognition was guided by anatomical knowledge, expressing its bipolarity. For instance, the putamen is exterior (i.e. to the right in the right hemisphere and to the left in the left one) of the union of lateral ventricles and thalamus (positive information) and cannot be interior (negative information); the putamen is quite close to the union of lateral ventricles and thalamus (positive information) and cannot be very far (negative information). Merging this information allows reducing the search area for the putamen, by dilating reference objects (lateral ventricles and thalamus in this example) with bipolar fuzzy sets representing these spatial constraints, thus focusing on the only regions of space where the spatial relations are satisfied, while avoiding forbidden regions.

Developing these preliminary examples, future work aims at applying this framework in the domain of spatial reasoning, in particular for knowledge-based object recognition in images. Another line of research is its application in the domain of preference modeling, for fusion, mediation and argumentation.

References

1. Dubois, D., Kaci, S., Prade, H.: Bipolarity in Reasoning and Decision, an Introduction. In: International Conference on Information Processing and Management of Uncertainty, IPMU 2004, Perugia, Italy, pp. 959–966 (2004)
2. Bloch, I.: Dilation and erosion of spatial bipolar fuzzy sets. In: Masulli, F., Mitra, S., Pasi, G. (eds.) WILF 2007. LNCS (LNAI), vol. 4578, pp. 385–393. Springer, Heidelberg (2007)
3. Bloch, I.: Bipolar fuzzy mathematical morphology for spatial reasoning. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 24–34. Springer, Heidelberg (2009)
4. Bloch, I.: Bipolar Fuzzy Spatial Information: Geometry, Morphology, Spatial Reasoning. In: Jeansoulin, R., Papini, O., Prade, H., Schockaert, S. (eds.) Methods for Handling Imperfect Spatial Information, pp. 75–102. Springer, Heidelberg (2010)
5. Bloch, I.: Lattices of fuzzy sets and bipolar fuzzy sets, and mathematical morphology. Information Sciences 181, 2002–2015 (2011)
6. Nachtegael, M., Sussner, P., Mélange, T., Kerre, E.: Some Aspects of Interval-Valued and Intuitionistic Fuzzy Mathematical Morphology. In: IPCV 2008 (2008)
7. Mélange, T., Nachtegael, M., Sussner, P., Kerre, E.: Basic Properties of the Interval-Valued Fuzzy Morphological Operators. In: IEEE World Congress on Computational Intelligence, WCCI 2010, Barcelona, Spain, pp. 822–829 (2010)
8. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, London (1982)
9. Ronse, C.: Why Mathematical Morphology Needs Complete Lattices. Signal Processing 21(2), 129–154 (1990)
10. Keshet, R.: Mathematical Morphology on Complete Semilattices and its Applications to Image Processing. Fundamenta Informaticae 41, 33–56 (2000)
11. Heijmans, H.J.A.M., Ronse, C.: The Algebraic Basis of Mathematical Morphology – Part I: Dilations and Erosions. Computer Vision, Graphics and Image Processing 50, 245–295 (1990)
12. Heijmans, H.: Morphological Image Operators. Academic Press, Boston (1994)
13. Grabisch, M., Greco, S., Pirlot, M.: Bipolar and bivariate models in multicriteria decision analysis: Descriptive and constructive approaches. International Journal of Intelligent Systems 23(9), 930–969 (2008)

14. Öztürk, M., Tsoukias, A.: Bipolar preference modeling and aggregation in decision support. International Journal of Intelligent Systems 23(9), 970–984 (2008)
15. Konieczny, S., Marquis, P., Besnard, P.: Bipolarity in bilattice logics. International Journal of Intelligent Systems 23(10), 1046–1061 (2008)
16. Dubois, D., Prade, H.: An Overview of the Asymmetric Bipolar Representation of Positive and Negative Information in Possibility Theory. Fuzzy Sets and Systems 160, 1355–1366 (2009)
17. Aptoula, E., Lefèvre, S.: A Comparative Study in Multivariate Mathematical Morphology. Pattern Recognition 40, 2914–2929 (2007)
18. Bouyssou, D., Dubois, D., Pirlot, M., Prade, H.: Concepts and Methods of Decision-Making. In: ISTE. Wiley, Chichester (2009)
19. Bloch, I.: Mathematical morphology on bipolar fuzzy sets: general algebraic framework. Technical Report 2010D024, Télécom ParisTech (November 2010)
20. Atanassov, K.T.: Intuitionistic Fuzzy Sets. Fuzzy Sets and Systems 20, 87–96 (1986)
21. Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning. Information Sciences 8, 199–249 (1975)
22. Neumaier, A.: Clouds, fuzzy sets, and probability intervals. Reliable Computing 10(4), 249–272 (2004)
23. Dubois, D., Gottwald, S., Hajek, P., Kacprzyk, J., Prade, H.: Terminology Difficulties in Fuzzy Set Theory – The Case of "Intuitionistic Fuzzy Sets". Fuzzy Sets and Systems 156, 485–491 (2005)
24. Deschrijver, G., Cornelis, C., Kerre, E.: On the Representation of Intuitionistic Fuzzy t-Norms and t-Conorms. IEEE Transactions on Fuzzy Systems 12(1), 45–61 (2004)
25. Bloch, I., Maître, H.: Fuzzy Mathematical Morphologies: A Comparative Study. Pattern Recognition 28(9), 1341–1387 (1995)
26. Bloch, I.: Duality vs. Adjunction for Fuzzy Mathematical Morphology and General Form of Fuzzy Erosions and Dilations. Fuzzy Sets and Systems 160, 1858–1867 (2009)

Image Decompositions and Transformations as Peaks and Wells

Fernand Meyer

Mines ParisTech,
Dpartement maths et systmes,
Centre de Morphologie Mathmatique,
F-77305 Fontainebleau Cedex, France

Abstract. An image may be decomposed as a difference between an image of peaks and an image of wells. This decomposition depends upon the point of view, an arbitrary set from where the image is considered: a peak appears as a peak if it is impossible to reach it starting from any position in the point of view without climbing. A well cannot be reached without descending. To any particular point of view corresponds a different decomposition. The decomposition is reversible. If one applies a morphological operator to the peaks and wells component before applying the inverse transform, one gets a new, transformed image.

1 Introduction

A binary image is made of particles and holes. Each particle may contain one or several holes and each hole one or several particles. These structures may be deeply nested. For describing this structure, J.Serra [8] introduced the homotopy tree, H.Heijmans [2] called it the adjacency tree. R. Keshet [3] and C. Ballester [1] studied it in depth and gave algorithms for constructing it. They then extended this tree construction to grey tone images, each in a different way, resulting in the so called tree of shapes.

In the present paper we propose a decomposition which decomposes any image into a peak and a well component, given a particular set, called point of view, an arbitrary set from where the image is considered: a peak appears as a peak if it is impossible to reach it starting from any position in the point of view without climbing. A well cannot be reached without descending. If the point of view intersect the minimum of a well, then this particular well will not be considered as well, since it is possible to reach any of its nodes without descending. To any particular point of view corresponds a different decomposition. Serra, Keshet of Ballester base their decomposition on a point in the background from which it is possible to apply hole-filling. Starting from an image X, a hole filling algorithm will produce a peak image P_1, then define the residue $R_1 = P_1 - X$, on which we can perform again a hole filling with a peak and residue, and so on until no residue is left. Summing up all peaks on one hand and all holes on the other hand produces the same result as our method if we adopt as view point. However, our method can use any set as view point and is also applicable to grey tone images.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 25–36, 2011.

Given a reference set X, called view-point set, we decompose an image in a difference between two components, one representing its peaks, the other its wells. They represent respectively the sums of positive and negative variations of the image if one follows a path starting in X. Ch.Ronse proposed an identical decomposition for functions of bounded variation on a poset P, with the final aim to find a sound way for constructing a flat operator on gray level image from a non-increasing operator on binary images [6].

The decomposition is reversible. If one applies a morphological operator to the peaks and wells component before applying the inverse transform, one gets a new, transformed image. In order to identify the admissible transforms on the peak and well components we study their algebraic structure, showing that they form a complete lattice. Openings, closings and morphological filters may then be derived from an adjunction defined on this lattice. Reconstruction openings, also called razings are also allowed opertors on this lattice. The last part of the paper is devoted to illustrations, showing how chosing an optimal point of view for the decomposition leads to interesting results. We conclude with a final discussion.

2 Decomposing an Image into Peaks and Wells

2.1 A Hiking Metaphor

A grey tone image may be considered as a topographic surface. Consider a hiker going from position x to position y on a montaineous landscape along a given route. Its tiredness will depend upon the total amount of climbing he has to do on his trip: no matter if he goes up and down, he only sums up the difference of level when he climbs. He may then chose the route along which this sum is minimal. Obviously, this minimal amount depends upon the starting point. If all possible starting points belong to a set X, then he may chose both the starting point x within X and the route between x and y which requires the minimum of climbing ; this minimal sum of climbing between X and y is a measure of the difficulty to reach y on the topographic surface if one starts from X.

Summing up only the differences of levels on the descending portions of the path would similarly yield a measure linked to the preceding measure: for a route between x and y, the altitude of y is equal to the sum of the altitude of x plus the total amount of climbing minus the total amount of descending. The next section gives a precise meaning to this hiking metaphor.

2.2 The Decompostion into Peaks and Wells

Let f be an image defined on a grid. This grid may be considered as a graph, where the pixels are the nodes and where two neighboring pixels i and j are linked by two arcs, one from i to j and the other from j to i. The weights of node i is f_i, the value of f at i. We consider three graphs e^+, e^- and $e^\#$ characterized by the following distribution of weights on the edges:

- $e^+ : f^+_{ij} = \vee(f_j - f_i, 0)$ positive for upwards transitions and null otherwise.
- $e^- : f^-_{ij} = \vee(f_i - f_j, 0)$ positive for downwards transitions and null otherwise.
- $e^\# : |f_{ij}| = |f_j - f_i|$

Consider now an arbitrary path $\pi = (x_1, x_2, ..., x_n)$ between two nodes x_1 and x_n. We may decompose $f(x)$ along the path π as follows:

$f(x_n) = f(x_n) - f(x_{n-1}) + f(x_{n-1}) - f(x_{n-2}) + ... + f(x_2) - f(x_1) + f(x_1)$

Since $f_j - f_i = f^+_{ij} - f^-_{ij}$ for neighboring pixels i, j in the path π, we get $f(x_n) = f(x_1) + \sum\limits_{ij \in \pi} f^+_{ij} - \sum\limits_{ij \in \pi} f^-_{ij}$.

Among all possible paths between x_1 and x_n, there is a path $\widehat{\pi}$ for which $\sum\limits_{ij \in \pi} f^-_{ij}$ is minimal. As $f(x_1) + \sum\limits_{ij \in \pi} f^+_{ij} - \sum\limits_{ij \in \pi} f^-_{ij}$ has a constant value, the expression $f(x_1) + \sum\limits_{ij \in \pi} f^+_{ij}$ is also minimal on $\widehat{\pi}$ and so is their sum $f(x_1) + \sum\limits_{ij \in \pi} f^+_{ij} + \sum\limits_{ij \in \pi} f^-_{ij} = f(x_1) + \sum\limits_{ij \in \pi} |f_{ij}|$.

The quantities $f(x_1) + \sum\limits_{ij \in \widehat{\pi}} f^+_{ij}, \sum\limits_{ij \in \widehat{\pi}} f^-_{ij}$ and $f(x_1) + \sum\limits_{ij \in \widehat{\pi}} |f_{ij}|$ respectively represent the positive, negative and total variation along $\widehat{\pi}$ in G.

The length of the shortest path $\widehat{\pi}(x_1, y)$ on e^+ between x_1 and any node y of G is a function $\Theta_{x_1}(e^+, f)$ which depends only upon x_1 and f and which takes the value $f(x_1) + \sum\limits_{ij \in \widehat{\pi}(x_1,y)} f^+_{ij}$ on the node y. Consider now multiple starting points of paths belonging to a set X. The minimum $\Theta_X(e^+, f) = \bigwedge\limits_{z \in X} \Theta_{x_1}(e^+, f)$ represents the peak component of each node ; for the node y it represents the minimal value taken by $f(z) + \sum\limits_{ij \in \widehat{\pi}(z,y)} f^+_{ij}$ for all paths starting at a node z in X and joining y.

We have seen that if $f(x_1) + \sum\limits_{ij \in \widehat{\pi}(x_1,y)} f^+_{ij}$ is minimal on the path $\widehat{\pi}(x_1, y)$, then $\sum\limits_{ij \in \widehat{\pi}(x_1,y)} f^-_{ij}$ and $f(x_1) + \sum\limits_{ij \in \pi} |f_{ij}|$ also are minimal on this same path. We obtain like that a function $\Theta_{x_1}(e^+, f)$ depending only upon x_1 and f and taking the value $\sum\limits_{ij \in \widehat{\pi}(x_1,y)} f^-_{ij}$ on the node y, representing the well component of y and a function $\Theta_{x_1}(e^\#, f)$ taking the value $f(x_1) + \sum\limits_{ij \in \widehat{\pi}(x_1,y)} |f_{ij}|$ representing the total variation at y.

As before we consider the minimum of these functions on all paths starting in X and joining y and define $\Theta_X(e^-, f) = \bigwedge\limits_{z \in X} \Theta_{x_1}(e^-, f)$ and $\Theta_X(e^\#, f) = \bigwedge\limits_{z \in X} \Theta_{x_1}(e^\#, f)$.

Finally, to each view point X corresponds a decomposition of the function f into a difference between a function representing the peaks and another representing the valleys: $f = \Theta_X(e^+, f) - \Theta_X(e^-, f)$. This relation also shows how f may be reconstructed from ist peak and well components.

2.3 Setting the Scenario of Trains Circulating on Graphs

This section explains how to construct these functions $\Theta_X(e^+, f)$ and $\Theta_X(e^-, f)$. $\Theta_X(e^+, f)(y)$ represents the minimal value taken by $f(z) + \sum_{ij \in \widehat{\pi}(z,y)} f_{ij}^+$ for all paths starting at a node z in X and joining y. This is an unconventional shortest path problem, as the length of the path $\pi = (x_1, x_2, ..., x_n)$ is equal to the sum of the arcs leading from x_1 to $x_{n,}$ plus the initial weight f_{x_1}. We again use a metaphor. The graph may be considered as a railway network, where the nodes are railway stations and the arrows are connections between them. Trains may follow all possible paths on G. However, they may only start from a subset $X \subset N$ of railway stations.

Consider a particular train. It starts at station $s \in X$ at time $\tau(s)$, follows a path $\theta = (x_0 = s, x_1, ..., x_n = t)$ where x_i and x_{i+1} are two railway stations linked by an arrow $(i, i+1)$ of E, weighted by the time $e_{i,i+1}$ needed for following it. The arrival time at destination is then $\tau(s) + \sum_{i,i+1 \in \theta} e_{i,i+1}$.

We now consider all trains starting at all possible nodes and taking all possible routes, and observe the earliest time when each node is reached by a train; if a train arrives at i before $\tau(i)$, we replace $\tau(i)$ by this first arrival time. For some nodes no train ever arrives ; for such a node i, all trains coming from another node arrive at i after $\tau(i)$. For others, no train ever departs: if $\tau(i)$ is too late, no train starting from i has a chance to be the first to reach another node ; this is in particular the case if $\tau(i) = \infty$. Some nodes cumulate both situations, and no train departs or arrives.

The resulting shedules $\widehat{\tau} = \Theta_X(e, \tau)$ depend on the distribution of initial departure times and crossing times of each edge. Defining τ_X^∞ by $\tau_X^\infty = \tau$ on X and $\tau_X^\infty = \infty$ elsewhere, it is obvious that $\Theta_X(e, \tau) = \Theta_X(e, \tau_X^\infty)$, since no train with an infinite departure time has the chance to reach another node first. $\Theta_X(e, \tau_X^\infty)$ is clearly an opening on the initial distribution of departure times on all nodes; it is obviously anti-extensive and increasing. It is also idempotent, as a second scheduling would not change the distribution $\tau(i)$ any further. We also remark that if $\tau(s) = 0$ for $s \in X$, the resulting schedules simply are the shortest path between X and all other nodes.

2.4 Harmonizing the Schedules Is a Shortest Path Problem in a Completed Graph

The preceding unconventional shortest path problem on the graph G can be transformed into a conventional one on an augmented graph G_X, obtained by adding to G a dummy node Ω with weight $\tau(\Omega) = 0$ and dummy edges (Ω, i) between Ω and each node i of X, with $e_{\Omega i} = \tau_i$. Since the travelling time along the edge (Ω, i) is τ_i, it is equivalent for a train to start from i at time τ_i or from Ω at time 0 and follow the edge (Ω, i) for reaching i.The earliest time for a train to arrive at any node k of G is the total duration of the shortest path between Ω and k. It follows that scheduling the graph G amounts to constructing the shortest path between Ω and all other nodes in the graph G_X, which is a classical problem in graph theory for which many algorithms exist. In "Scheduling trains with delayed departures" (http://hal.archives-ouvertes.fr/hal-00547261/fr/), we have presented the algorithms of Moore Dijkstra and of Berge and shown that, if the introduction of a dummy node Ω and dummy edges (Ω, i) is a useful support for thinking, it is not necessary in practice.

3 The Lattice of Floodings with a Unique Regional Minimum

We are now able to decompose a function f into its peak and wells components, given the point of view set X. The function f may be reconstructed by $f = \Theta_X(e^+, f) - \Theta_X(e^-, f)$. Applying an operator ψ to each component leads to a transform $\widehat{\psi}(f) = \psi\Theta_X(e^+, f) - \psi\Theta_X(e^-, f)$. This section analyses the algebraic structure of the peak and wells components in order to identify which operators ψ are admissible and transform a peak or wells component into an other grey tone image with the same characteristics.

The schedule $\widehat{\tau_X}(x) = \Theta_X(e, \tau)$ is the length of the shortest path between Ω and the node x. The edge weights being non negative, there exists a non ascending path between any node and Ω for the valuations $\widehat{\tau_X}$. Hence Ω is the only regional minimum of $\widehat{\tau_X}$.

The following formulations are equivalent:

- Ω is the only regional minimum of $\widehat{\tau_X}$.
- the only regional minima of τ belong to X.
- $\widehat{\tau_X}$ is invariant by the swamping which imposes Ω as only regional minimum ; that is the reconstruction closing of $\widehat{\tau_X}$ with a marker function 0_Ω^∞ equal to 0 at Ω and ∞ elsewhere. This operation, also called flooding, is an anti-extensive leveling [5], written $\Lambda^+(\widehat{\tau_X}, 0_\Omega^\infty)$.
- $\widehat{\tau}$ is invariant by the swamping which imposes the minima of X as only regional minima ; that is the flooding $\Lambda^+(\widehat{\tau}, \tau_X^\infty)$.
- considering the threshold at valuation λ, the subgraph spanning all nodes with a valuation $\tau < \lambda$ (we call it the background at level λ) has only one connected component, and this component contains Ω.

We call F_X the lattice of all functions verifying the previous equivalent criteria.

3.1 Infimum and Supremum in the Lattice F_X

F_X is a complete lattice, with the ordinary order relations for functions, 0 as minimal element, ∞ as maximal element.

If h_1 and h_2 are two functions of F_X, their infimum in F_X is simply $h_1 \wedge h_2$. As a matter of fact, the bakgrounds of h_1 and h_2 at level λ comprise each one connected component ; as they both contain Ω, their union, which is the background of $h_1 \wedge h_2$ also consists in one connected component containing Ω.

Now, $h_1 \vee h_2$ may have regional minima outside X, which have to be suppressed by swamping. For this reason the supremum $h_1 \sqcup h_2$ is equal to $\Lambda^+(h_1 \vee h_2;\ (h_1 \vee h_2)_X^\infty)$.

Remark: These operators are not distributive one with another.

3.2 An Adjunction in the Lattice F_X

If for each node x we define a neighborhood V_x, the erosion of a function f with this variable structuring element is $\varepsilon_V f(x) = \bigwedge_{y \in V_x} f_y$. This operator is an erosion both in the ordinary lattice of functions as in the lattice F_X. Its adjunct dilation in the ordinary lattice is $\delta_V f(y) = \bigvee_{x \in V_y^t} f_x$, where $V_y^t = \{x \mid y \in V_x\}$. Being an adjunction, they verify for each couple (f, g) of functions the relation: $f < \varepsilon_V g \Leftrightarrow \delta_V f < g$.
As $\delta_V f(y) = \bigvee_{x \in V_y^t} f_x$ does generally not belong to F_X, we construct $\widetilde{\delta_V} f(y) = \bigsqcup_{x \in V_y^-} f_x = \Lambda^+(\delta_V f, (\delta_V f)_X^\infty)$. To establish that $\widetilde{\delta_V}$ is the ajunct of ε_V in F_X, we have to prove that $\delta_V f < g \Leftrightarrow \Lambda^+(\delta_V f, (\delta_V f)_X^\infty) < g$.

On one hand, floodings being extensive we have $\Lambda^+(\delta_V f, (\delta_V f)_X) < g \Rightarrow \delta_V f < g$.
On the other hand, if f and g belong to $F_{X.}$, then $\delta_V f < g \Rightarrow \Lambda^+(\delta_V f, (\delta_V f)_X) < \Lambda^+(g, g_X)$. But $\Lambda^+(g, g_X) = g$, as g is invariant by the flooding $\Lambda^+(g, g_X)$.
Concatenating all equivalences, we get for any couple of functions in F_X: $f < \varepsilon_V g \Leftrightarrow \delta_V f < g \Leftrightarrow \widetilde{\delta_V} f = \Lambda^+(\delta_V f, (\delta_V f)_X) < g$, showing that $(\varepsilon_V f, \widetilde{\delta_V} f) = (\bigwedge_{y \in V_x} f_y, \bigsqcup_{x \in V_y^-} f_x)$ is indeed an adjunction on F_X.

Classically $\varepsilon_V \widetilde{\delta_V}$ is then a closing and $\widetilde{\delta_V} f \varepsilon_V$ an opening. Based on these openings one may construct all classical morphological filters based on openings and closings [4],[9].

Razings clip peaks and do not create new holes. Thus the image of F_X by any type of razing or reconstruction opening is F_X. Finally, we obtain a large family of operators which operate on and in F_X.

4 Min, Max and Morphological Operators Through the Decomposition in Peaks and Wells

Suming up, given a set X, serving as view-point, we are now able to decompose any image f into a cumulative image of its peaks f_X^+ and a cumulative image of its wells f_X^-, such that $f = f_X^+ - f_X^-$. Both functions belong to F_X , a lattice where we defined a supremum, infimum and an adjunction. Furthermore this lattice is stable by any type of razing.

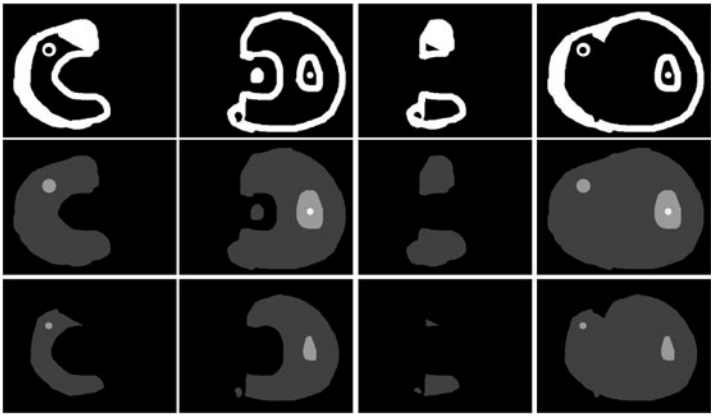


Fig. 1.
Columns 1 and 2 : Binary sets Z and Y and their decomposition
Column 3 : $Z \curlywedge Y$ and its decomposition
Column 4 : $Z \curlyvee Y$ and its decomposition

4.1 "Supremum" and "Infimum" of Two Functions

Let $(f_X^+,\ f_X^-)$ and (g_X^+, g_X^-) be the decomposition of two function into peaks and wells, given the view point X. For any composition law $\triangle$ on F_X, we may define $f \triangle g = f_X^+ \triangle g_X^+ - f_X^- \triangle g_X^-$.

Remark: We do not necessarily have $(f \triangle g)_X^+ = f_X^+ \triangle g_X^+$ and $(f \triangle g)_X^- = f_X^- \triangle g_X^-$. This means that the decomposition $(f_X^+ \triangle g_X^+, f_X^- \triangle g_X^-)$ is not necessarily a minimal decomposition of $f \triangle g$.

Based on the operator $\wedge$ and $\sqcup$ in F_X, we obtain $f \curlywedge g = f_X^+ \wedge g_X^+ - f_X^- \wedge g_X^-$ and $f \curlyvee g = f_X^+ \sqcup g_X^+ - f_X^- \sqcup g_X^-$.

Both operators are illustrated by the images I_{rc} disposed in a matrix in fig.1. Row 1 of the matrix shows the binary images, row 2 and 3 respectively their peak and wells components, taking as point of view set X the boundary of the image. Column1 and 2 show two sets Z and Y, column 3 their intersection $Z \curlywedge Y$ and column 4 their union $Z \curlyvee Y$. The images I_{23} and I_{24} represent respectively $Z_X^+ \wedge Y_X^+$ and $Z_X^+ \sqcup Y_X^+$. The images I_{33} and I_{34} represent respectively $Z_X^- \wedge Y_X^-$ and $Z_X^- \sqcup Y_X^-$. The images I_{13} and I_{14} represent respectively $Z \curlywedge Y = Z_X^+ \wedge Y_X^+ - Z_X^- \wedge Y_X^-$ and $Z \curlyvee Y = Z_X^+ \sqcup Y_X^+ - Z_X^- \sqcup Y_X^-$.

4.2 Operators Operating on the Peaks and Wells

Binary images. Given the decomposition $(f_X^+,\ f_X^-)$ of a function f into peaks and wells, and given an increasing operator ψ from F_X into F_X, we define $\widehat{\psi} f = \psi\left(f_X^+\right) - \psi\left(f_X^-\right)$. The decomposition is not necessarily minimal, but since $f_X^- < f_X^+$ and ψ increasing, $\widehat{\psi} f$ will be positive if f is positive.

Figure 2 presents in a matrix a number of operators on a binary image. The view point set X is the boundary of the image. The first row shows successively

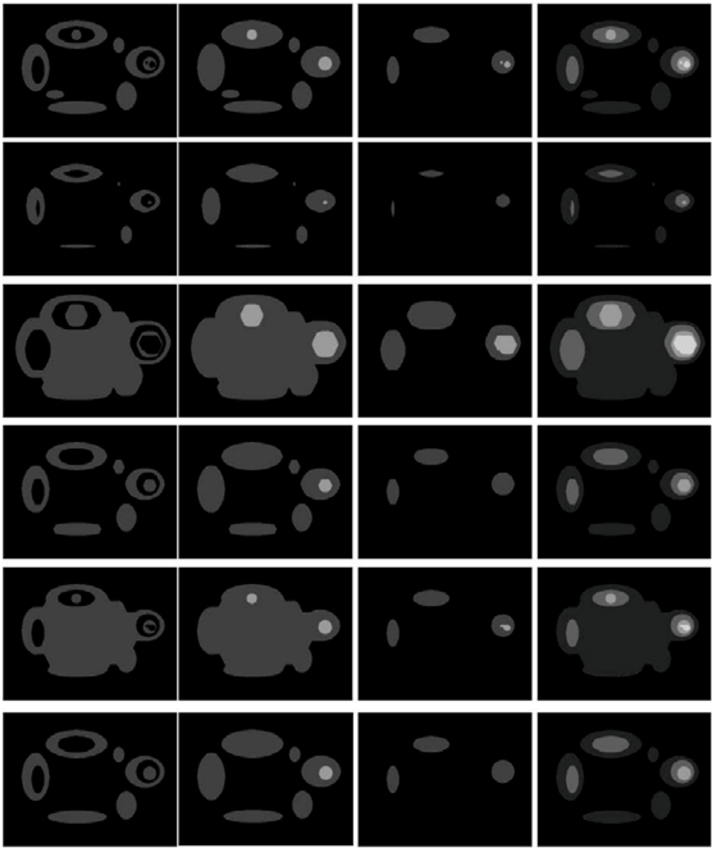


Fig. 2. Decomposition and transformations on binary images

the initial image f, its peak component f_X^+, its wells component f_X^-, and the total variation $f_X^+ + f_X^-$. Rows 2-6 present each a different tranform $\widehat{\psi} f = \psi\left(f_X^+\right) - \psi\left(f_X^-\right)$. First $\psi\left(f_X^+\right)$ and $\psi\left(f_X^-\right)$, represented in columns 2 and 3 are constructed, then $\psi\left(f_X^+\right) - \psi\left(f_X^-\right)$ represented in column 1 and $\psi\left(f_X^+\right) + \psi\left(f_X^-\right)$ represented in column 4. The transforms illustrated in each row are: 1) initial image; 2) erosion of size 11; 3) dilation of size 15; 4) opening of size 11; 5) closing of size 15; 6) reconstruction opening of size 11.

The matrix of images I_{rc} in figure 3 also presents an erosion of size 11 of the same binary image, but the decomposition is made with respect to a different view point set X, represented as a white dot superimposition to the binary set f in image I_{11}. X is contained in a white particle of Z, itself contained in a hole contained in a particle. Images I_{12} and I_{13} present the peaks and wells components of Z with respect to the point of view X, and image I_{13} the total variation. An erosion ε_{11} of size 11 is applied to the peak and wells components represented in images I_{22} and I_{23}. The transformed image $\widehat{\varepsilon_{11}}\,(f) = \varepsilon_{11}\left(f_X^+\right) - \varepsilon_{11}\left(f_X^-\right)$ is presented in image I_{21} and the total variation $\varepsilon_{11}\left(f_X^+\right) + \varepsilon_{11}\left(f_X^-\right)$ in figure I_{24}.

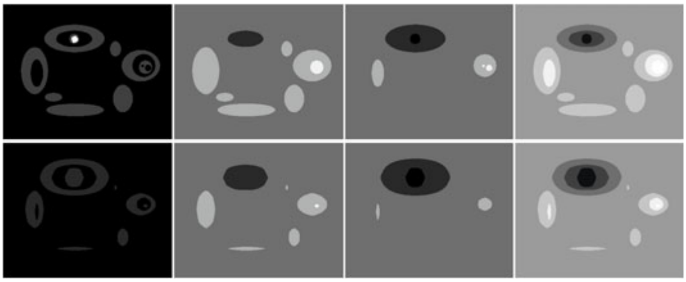


Fig. 3. Decomposition and transformations on binary images. The view point set is the white particle in the first image.

The second row of both figures 2 and 3 present an erosion of size 11, but with respect to a different point of view set X. The first example takes for X the boundary of the image, the second a dot contained within in a particle. In this last case, particles containing X are treated differently from particles which do not contain it. As the particles and holes containing X appear as holes both in the peak as in the wells image, their erosion enlarges them. On the contrary, particles and holes not containing X appear as peaks and are indeed eroded.

Grey tone images. Fig.4 presents the decomposition and transformation of grey tone images, if one takes as point of view the boundary of the image ; there are four matrices of images with an identical disposition. In each matrix the first row shows the positive variation σ^+ and negative variation σ^-, whereas the second row shows the initial image and the total variation $\sigma^\#$. The first matrix presents the initial image and its decomposition. The second matrix an erosion of size 3, the third a dilation of size 3 and the last an opening of size 3. Since the first letter "P" of "Paulus" intersects the boundary of the image, which plays the role of view point set X, it is not transformed as the other letters.

The last example in fig. 5 is taken from an image of the retina. The aim is to detect the small dots and the bloodings without detecting the vessel, although they have the same width. The images are organized as two matrices with the same structure. Both matrices differ by the choice of the view point set. For the left matrix the view point set is the left boundary of the image, and for the right matrix the white segment in superimposition with the vessel in image I_{11}. Both matrices are organized as follows. I_{11} the initial image, I_{12} its filtered version which is decomposed. The peaks are in I_{21}, the wells in I_{31}.

Consider the vessel which crosses the image. In the left matrix, where the left boundary of the image is the point of view set, the right boundary of the vessel produces an upwards transition in the peaks component in I_{21}. In the wells component in I_{31} it is the left boundary of the vessel which produces an upwards transition. Like that, with an appropriate choice of a point of view set, a thin dark structure like the vessel appears in each component as a step for climbing on a plateau. Small dark and round structures, here microaneurisms appear indeed as bright peaks in the wells component and not at all in the peaks component.

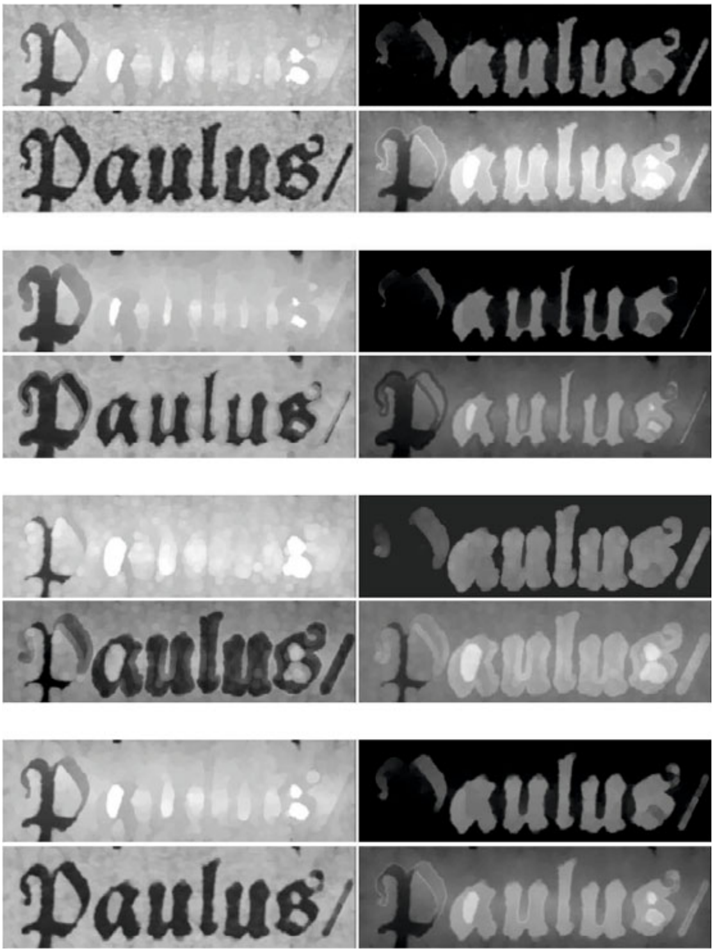


Fig. 4. Decomposition and transformation of grey tone images.

A reconstruction opening after a small erosion of both components produces duly suppresses the aneuvrism but leaves the step of the plateau unchanged (see images I_{22} and I_{32}). Subtracting the initial image in I_{12} from the reconstructed image after reconstruction opening of the components in I_{41} yields the final result shown in I_{42}.

Consider now the right matrix. As the point of view is inside the vessel, the vessel does not appear as peak in the wells components in I_{31}. On the contrary in the peaks component I_{21}, it appears as a well. Here again the behaviour of the aneurisms are different, they are not visible in the peaks component and appear as peaks in the wells component. An erosion applied to the components blows up the vessel in the peaks component in I_{22} and suppresses the aneurisms in the wells component. The recomposed image after erosion in I_{41} shows a blown up vessel and no aneurisms. Subtracting $I_{12} \vee I_{41}$ from the initial image I_{12} yields

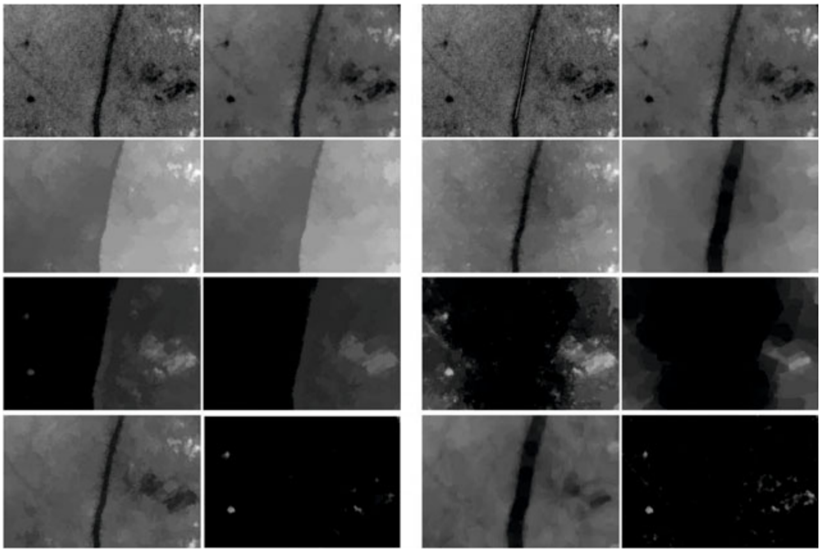


Fig. 5. Analysis of the retina in order to detect the micro-anevrysms and discard the vessel

a residual image in I_{42} where the vessel has disappeared and only the aneurisms are visible.

5 Conclusion

Decomposing an image into peaks and wells leads to interesting operators. We have several degrees of freedom. First chose the right point of view. Then apply the adequate operator on the peaks and wells components. The analysis of the lattice structure of the wells and peaks like images has shown that a large collection of operators are available. The construction of the decomposition is fast, as it only involves shortest paths algorithms. As the resulting component images turn the peaks and wells of the original image into peaks, they may advantageously be encoded as max-trees [7] in order to enlarge the scope of applicable operators to the peaks and wells components and increase the speed of processing.

It is also interesting to decompose the same image with respect to several distinct points of view. Changing the point of view produces a different decomposition of the same scene, with a different distribution of contrast, between the same objects, which may facilitate the segmentation.

More exotic variants can be envisaged. Changing the edge weights, for instance by keeping only the variations along each edge which are higher than a threshold will put the focus with contrasted contours and ignore the objects with a more fuzzy contour. The distribution of weights on the edges may also be directional, favoring some directions and discarding others. This gives the possibility to analyze anisotropies.

References

1. Ballester, C., Caselles, V., Monasse, P.: The tree of shapes of an image. ESAIM: COCV 9, 1–18 (2003)
2. Heijmans, H.J.A.M.: Connected morphological operators for binary images. Computer Vision and Image Understanding 73(1), 99–120 (1999)
3. Keshet, R.: Adjacency lattices and shape-tree semilattices. Image and Vision Computing 25(4), 436–446 (2007); International Symposium on Mathematical Morphology 2005
4. Matheron, G.: Filters and lattices. In: Serra, J. (ed.) Mathematical Morphology. Theoretical Advances, vol. II. Academic Press, London (1988)
5. Meyer, F.: The levelings. In: Heijmans, H., Roerdink, J. (eds.) Mathematical Morphology and Its Applications to Image Processing, pp. 199–207. Kluwer, Dordrecht (1998)
6. Ronse, C.: Bounded variation in posets, with applications in morphological image processing. In: Proceedings of the Kiselmanfest, Acta Universitatis Upsaliensis, vol. 86, pp. 249–283 (2009)
7. Salembier, P., Garrido, L.: Connected operators based on region-tree pruning. In: Goutsias, J., Vincent, L., Bloomberg, D.S. (eds.) Mathematical Morphology and its Applications to Image and Signal Processing, vol. 18, pp. 169–178. Springer, US (2002)
8. Serra, J.: Mathematical Morphology, vol. I. Academic Press, London (1982)
9. Serra, J.: Alternating Sequential Filters. In: Mathematical Morphology, vol. II. Academic Press, London (1988)

Grain Building Ordering

Jean Serra

Université Paris-Est, 2, Bd Blaise Pascal, 93162l Noisy-le-Grand, France
j.serra@esiee.fr

Abstract. Given a set E, the partitions of E are usually ordered by merging of classes. In segmentation procedures, this ordering often generates small parasite classes. A new ordering, called "grain building ordering", or GBO, is proposed. It requires a connection over E and states that $A \preccurlyeq B$, with $A, B \subseteq E$, when each connected component of B contains a connected component of A. TheGBO applies to sets, partitions, and numerical functions. Thickenings ψ with respect to the GBO are introduced as extensive idempotent operators that do not create connected components. The composition product $\psi\gamma$ of a connected opening by a thickening is still a thickening. Moreover, when $\{\gamma_i, i \in I\}$ is a granulometric family, then the two sequences $\{\psi\gamma_i, i \in I\}$ and $\{\gamma_i\psi, i \in I\}$ generate hierarchies, from which semi-groups can be derived. In addition, the approach allows us to combine any set of partitions or of tessellations into a synthetic one.

1 Introduction

In image processing, the segmentation techniques, which aim to partition the space of definition of the image under study, often generate a few correct classes. They are large and representative, but surrounded by a multitude of small parasitic other ones. Figure 1, depicts a typical example of the phenomenon. Several authors, such as Ph. Salembier et Al. [9] or J. Crespo et Al. [3], among others, propose solutions by merging of flat zones that satisfy convenient criteria, in association with some constraints (e.g. not to subdivide the small zones). In [14], P. Soille and J. Grazzini try to stamp out the parasites by imposing the presence of one extremum at least of the image inside each segmented class. In [1], the small regions are removed by erosions of partitions.

Small classes turn out to be inherent in the segmenting techniques. In case of connective segmentation, for example, they satisfy the chosen criterion, in the same way as the large classes. If they are reduced by intersection of constraints, they become singletons [11], or they are absorbed by the background [7]. The trouble is shifted, but not solved: what to do, then, with this background, or with these unclassified singletons? The solution adopted in [11] (figures 11 and 12) consists in building the influence zones of the large classes, which absorb the small ones. Independently, the above authors often did the same, but surreptitiously, for not catching attention on a procedure that resembles to cooking, rather than to nice theorems.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 37–48, 2011.

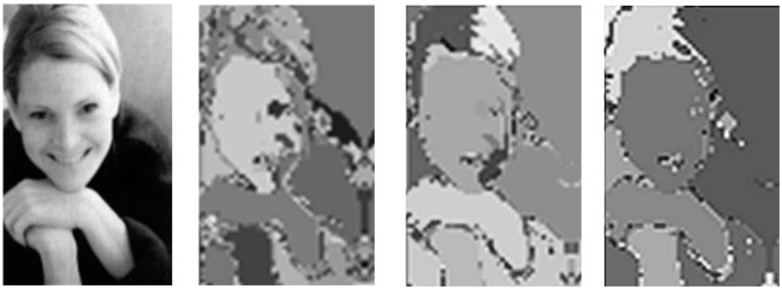


Fig. 1. Segmentations quasi-flat zones of increasing slopes λ; as λ increases, the details of the face progressively vanish, though the parasite small grains remain (by courtesy of Noyel et Al. [6])

The elimination of the small classes refers to a wider question: what does the usual ordering on partitions stand for? According to this so called "refinement ordering", one goes from a smaller partition to a larger one uniquely by merging classes, hence by only removing frontiers and not by moving them, or by creating new ones. This rapidly leads to ambiguous situations whose paradigm is depicted in figure 2. In case of the figure, must we introduce some choice in a segmenting approach which is basically deterministic?

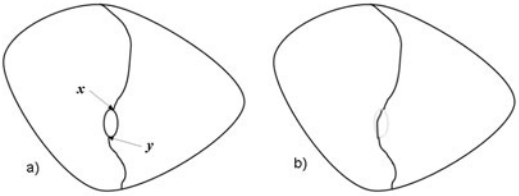


Fig. 2. The usual ordering on partitions can suppress the small class in a) only by merging it with one of the two large ones. Could another ordering divide up the small central class among the two others, as in b)?

When passengers are fully packed in the metro, and that one person leaves the carriage, does one of his neighbors suddenly swell and monopolize the whole free space for himself? At the end of a war, do both winners and losers decide, as an absolute rule, not to move any frontier? In figure 2 as well, we would like to refuse the ukase of the refinement ordering, and to share the small class among the two large ones, by joining x to y by a simple arc. If we proceed this way, then every class of the resulting partition contains at least one class of the initial one. It is exactly this property that we will now raise to an axiom[1].

[1] A first version of this work, with all proofs and supplementary results, was presented in a worksop at ESIEE, in April 2010 [13]. This initial work inspired Ch.Ronse with several other orderings on partial partitions, and with the resulting optimisations. They can be found in these proceedings [8].

2 Grain Building Ordering (GBO)

Given set E, we consider the lattice $\mathcal{P}(E)$ of all its subsets, and provide it with an arbitrary connection $\mathcal{C}^*$, said to be standard. Unlike inclusion, where $A \subseteq B$ means that any *point* of A belongs to B, the ordering relation introduced below holds on the $\mathcal{C}^*$*-components de* $\mathcal{P}(E)$, hence its generic appellation of connected ordering. We will indicate a few notation. The image of $\mathcal{P}(E)$ under operation ψ is written by $\mathcal{P}_\psi$

$$\mathcal{P}_\psi = \{\psi(X), X \in \mathcal{P}(E)\}.$$

In order to avoid confusion between the various openings that intervene, we denote by $\dot{A}_x$, the $\mathcal{C}^*$-component of A at point x (instead of $\gamma_x(A)$). When the labelling of the $\mathcal{C}^*$-components of A is not necessary, one just writes $\dot{A}$ (with $\dot{A} \subseteq A$). When the context is not ambiguous the expression ”C^*- connected component” is replaced by ”component”, or by ”grain”, and ” C^*-connected opening” by ”connected opening”. On the other hand, we keep the same symbol $\subseteq$ to designate the setwise inclusion as well as the ordering relation it induces between operators (i.e. $\gamma \subseteq \gamma'$ iff $\gamma(A) \subseteq \gamma'(A)$ for all $A \subseteq E$).

Let $A \in \mathcal{P}(E)$, of connected components $\dot{A}_i$. An anti-extensive grain operator is an operation on A that suppresses some $\dot{A}_i$ and leave unchanged the others, according to an increasing binary criterion that holds on each grain separately. When the grain operator γ is idempotent, we speak of connected opening. It may be the matter of an area threshold, or of the radius of the inscribed disc, or of any external attribute.

An extensive and idempotent operation $\psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$, is traditionally called thickening, or sometimes idempotent thickening [10]. In addition we assume here that the thickening ψ does not create connected components. Given connection C^*, neither the grain operators and their connected openings, nor thickenings, involve the lattice structure of $\mathcal{P}(E)$, but its inclusion ordering only. Therefore, these three notions apply for any partial ordering on $\mathcal{P}(E)$.

Proposition 1. *When $\mathcal{P}(E)$ is equipped with connection $\mathcal{C}^*$, then the relation*

$$A \preccurlyeq B \textit{ iff each } \dot{B} \subseteq B \textit{ contains at least one } \dot{A} \tag{1}$$

defines over $\mathcal{P}(E)$ a $\mathcal{C}^*$-connected ordering, called grain building ordering *(in short: GBO).*

For example, a $\subseteq$-thickening ψ that does not create connected components is not only $\subseteq$-extensive, but also $\preccurlyeq$-extensive, and as its idempotence is independent of the ordering, it turns out to be a $\preccurlyeq$-thickening as well. It is the same for any connected opening γ. We have indeed that $\gamma(A) \succeq A$, since every $\mathcal{C}^*$-component of $\gamma(A)$ contains a $\mathcal{C}^*$-component of A, namely itself. Note that the $\preccurlyeq$-order of $\gamma(A)$ and A is the opposite of that of inclusion. Remark also that the axiom of the conditional union, which characterizes the connection, does not intervene in Proposition 1.

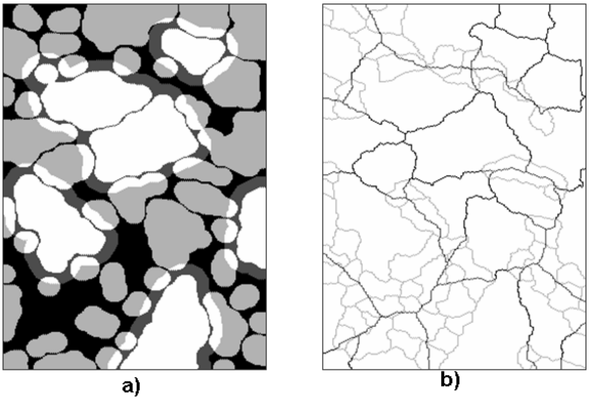


Fig. 3. GBO: a) for sets: the dilates of a few grains (in white) generate a set larger than the initial one; b) for partitions: in dark grey, the larger partition

Every set ordering relation extends to partitions via their classes. In the present case, we can state the following:

Corollary 2. *The set property (1) generates an ordering on the space $\mathcal{D}$ of the partitions of E with $\mathcal{C}^*$-connected classes, where $D_1 \preccurlyeq D_1$, $D_1, D_2 \in \mathcal{D}$, when each class of D_2 contains one class of D_1 at least.*

Clearly, the set GBO does not preserve inclusion, neither connection $\mathcal{C}^*$, since when $A \preccurlyeq B$ some $\mathcal{C}^*$-components of the smaller set, A, may lie partly or even completely outside of B, as shown in figure 3. In case of sets, GBO is thus non comparable to inclusion, though, for partitions, it is more restrictive than the usual refinement:

$$D_1 \leq D_2 \quad \Rightarrow \quad D_1 \preccurlyeq D_2, \quad D_1, D_2 \in \mathcal{D}, \tag{2}$$

an implication which is no longer true for partial partitions. For extending the setwise GBO to functions, one proceeds by comparing each section $X_f(t) = \{x : x \in E, f(x) \geq t\}$ to the analogue $X_g(t)$, by putting

$$f \preccurlyeq g \;\; \Leftrightarrow \;\; X_f(t) \preccurlyeq X_g(t), \;\; \forall t \in \bar{\mathbb{R}} \text{ (or } \forall t \in \overline{\mathbb{Z}}) \; .$$

which defines a numerical ordering, which is illustrated by Figure 4.

The set GBO does not induce any lattice: if $\dot{A}_1$ and $\dot{A}_2$ are two $\preccurlyeq -$components of A, each of both sets $\dot{A}_1 = A_1$ and $\dot{A}_2 = A_2$ is an upper bound of A, but there is no upper bound of A smaller than A_1 and A_2. Therefore one can still introduce increasing mappings.

By duality under complementation, Relation (1) induces the following one:

$$A^\star \preccurlyeq B^\star \text{ if every } (\dot{B}^c) \subseteq B^c \text{ contains at least one} (\dot{A}^c) \subseteq A^c, \tag{3}$$

which is still an ordering relation. The two relations (1) and (3) are not equivalent, neither incompatible, and their logical intersection defines a third ordering,

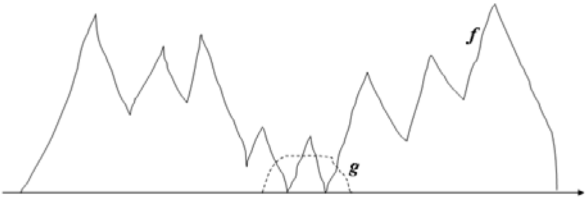


Fig. 4. GBO for numerical functions. Function g, in dotted lines, is the greater.

of type homotopic type, in that it makes symmetrical the roles of grains and pores. Other orderings associated with connections and partitions can be defined [13] [8]. In particular, one can take the logical intersection between GBO and inclusion, which eliminates all outside small objects. This yields the restriction of the GBO to the partial partitions with same support. The $\preccurlyeq$-thickenings below satisfy this double ordering.

This partial GBO governs the variations of some physical phenomena, such as changes of metallic grains under fatigue. It appears also in political changes (e.g. the dismemberment or the Ottoman Empire at the end of 19th century).

3 Thickening and $\preccurlyeq$ Ordering

3.1 $\preccurlyeq$-Thickening from Connected Opening

We will now construct operations that simplify sets and partitions, by sorting out certain main regions which then expand and cover the whole space. Their choice is governed by an opening, and their expansion by a thickening. The simplest, but the most worked out case, occurs when the opening is $\mathcal{C}^*$-connected:*

Proposition 3. *Given a connection $\mathcal{C}^*$ on $\mathcal{P}(E)$, let $\gamma : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ be a $\subseteq$-anti-extensive grain operator, and $\psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ be a $\subseteq$-thickening that does not create connected components. The composition product $\psi\gamma$ is a thickening for the GBO, and we have*

$$I \preccurlyeq \psi\gamma = \gamma\psi\gamma = (\psi\gamma)^2. \tag{4}$$

The product $\gamma\psi$ also behaves as a thickening, up to factor ψ:

$$I \preccurlyeq \psi\gamma\psi = \gamma\psi\gamma\psi = (\gamma\psi)^n \qquad n > 1. \tag{5}$$

Below, the grain operators of the proposition are always connected openings. Note that the proposition does not inform us on the distance between connected components in $\psi\gamma$ and $\gamma\psi$.

Extension to partitions. Proposition 3 can be stated in terms of partitions of E into connected classes. Let $\mathcal{D}$ be the set of these partitions, $D \in \mathcal{D}$ and D_x the class of D at point x. The set connected opening γ induces on $\mathcal{D}$ the following operation $\gamma_{\mathcal{D}}$

$$D_x[\gamma_{\mathcal{D}}(D)] = \gamma(D_x) = D_x \quad \text{if} \quad x \in \gamma(D_x)$$
$$D_x[\gamma_{\mathcal{D}}(D)] = \{x\} \quad \text{if not}$$

Proposition 3 extends to partitions by replacing $\gamma : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ by $\gamma_{\mathcal{D}} : \mathcal{D}(E) \rightarrow \mathcal{D}(E)$, and by using a thickening $\psi : \mathcal{D}(E) \rightarrow \mathcal{D}(E)$.

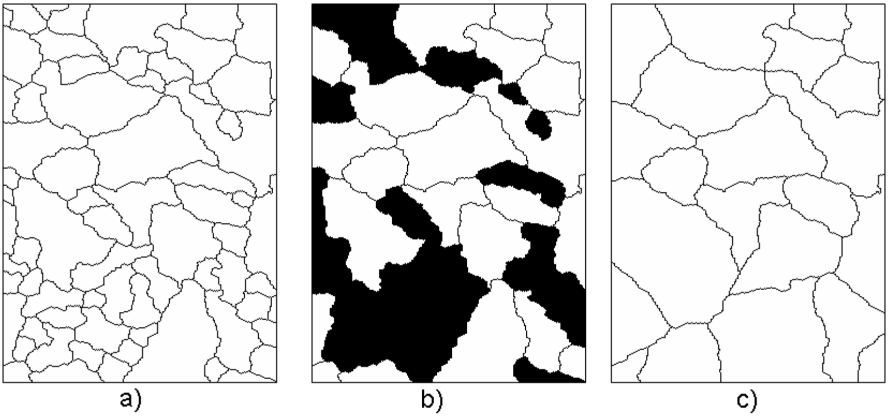


Fig. 5. a) Initial tessellation A, b) opening $\gamma(A)$ that suppresses grains according to their inscribed disc (here for radius ≤ 15); c) Voronoi thickening $\psi\gamma(A)$ of $\gamma(A)$, which is identical to its opening $\gamma\psi\gamma(A)$

$\preccurlyeq$-thickening from non-connected opening. When opening γ is not connected, then Proposition 3 is no longer valid, and is replaced by a more specific result.

Proposition 4. *Let γ be an opening on $\mathcal{P}(E)$ that acts independently on connected components, and let $\psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ be a $\subseteq$-extensive operator that does not create $\mathcal{C}^*$- components. Denote by $(\widetilde{\gamma\psi})\gamma(A)$ the union of those $\mathcal{C}^*$-components of $\gamma\psi\gamma(A)$ that contain a $\mathcal{C}^*$-component of $\gamma(A)$. The composition product $(\widetilde{\gamma\psi})$ is then $\preccurlyeq$-extensive on $\mathcal{P}_\gamma = \gamma[\mathcal{P}(E)]$:*

$$\gamma(A) \preccurlyeq (\widetilde{\gamma\psi})^j\gamma(A) \preccurlyeq (\widetilde{\gamma\psi})^{j+1}\gamma(A). \tag{6}$$

The idempotence of ψ is not necessary, and the condition, on γ, of individual processing is satisfied by the usual openings by convex structuring elements. For finite sets of $E = \mathbb{Z}^2$ the limit $\mu = (\widetilde{\gamma\psi})^n\gamma = (\widetilde{\gamma\psi})^{n+1}\gamma$ is reached after n steps, $n < \infty$ (see Figure 6).

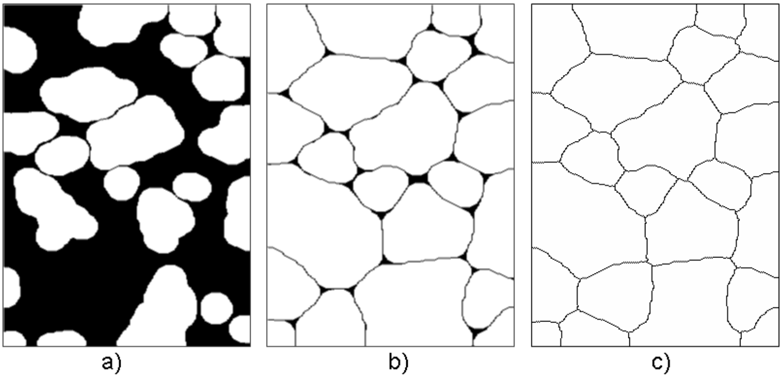


Fig. 6. The initial tesselation is that of figure 7; a) opening of the classes by a dodecagon of size 15; b) limit opening μ, c) limit Voronoi thickening $\psi\mu$ (and $\gamma\psi\mu = \mu$)

Tessellations, partitions and Voronoi thickenings

Tesselations. In $\mathbb{R}^2$, it is convenient to distinguish between a partition and the opening of its classes. Following R. Miles, we shall call "tesselation" any set of $\mathbb{R}^2$ whose all $\mathcal{C}^*$-components but one are topologically open, the last one being a locally finite union of simple arcs. These contours are called "cleavages", and the open classes "tassels" [13]. When the cleavages class is connected, then the tassels are simply connected. The practical interest of a tessellation is that its open classes can always be handled as subsets of $\mathcal{P}(\mathbb{R}^2)$.

Voronoi thickening in $\mathbb{R}^2$*.* Start from the family $\mathcal{G}_0$ of all locally finite unions of disjoints open sets. Let $A = \cup\dot{A}_k \in \mathcal{G}_0 \subseteq \mathcal{P}(\mathbb{R}^2)$. The *zone of influence of* $\dot{A}_k$ is the set of all points closer to $\dot{A}_k$ than to any other $\dot{A}_p \in A, p \neq k$, and the *Voronoi thickening* of A is the union $\psi(A)$ of all zones of influence. The complement set $[\psi(A)]^c$ is a locally finite union of simple arcs [4], called skeleton by zones of influence. Therefore, the operator ψ is a $\subseteq$-thickening on $\mathcal{G}_0$ that does not create connected components, hence a $\preccurlyeq$-thickening. Consider now a grain opening $\gamma : \mathcal{G}_0 \rightarrow \mathcal{G}_0$, then Rel. (4) implies that the composition product $\psi\gamma : \mathcal{G}_0 \rightarrow \mathcal{G}_0$ is still a $\preccurlyeq$-thickening.

Although ψ is not $\preccurlyeq$-increasing in general, it becomes $\preccurlyeq$-increasing for those pairs A and $A' \in \mathcal{G}_0$, such that $A' = A \cup B$, $B \in \mathcal{G}_0$, the $\mathcal{C}^*$-components of B being disjoint from those of A.

Proposition 5. *Let* $A, A', B \in \mathcal{G}_0$*, with* $A' = A \cup B$*, and* $B \cap A = \varnothing$*. Then the Voronoi thickening* ψ *is* $\preccurlyeq$*-increasing, i.e.*

$$\{A' = A \cup B, \quad B \cap A = \varnothing\} \quad \Rightarrow \quad \{A' \preccurlyeq A \quad \Rightarrow \quad \psi(A') \preccurlyeq \psi(A)\}. \qquad (7)$$

Voronoi thickening in $\mathbb{Z}^2$*.* One cannot transpose the above approach directly to $\mathbb{Z}^2$, because the involved digital distances do not ensure that the connectivity of

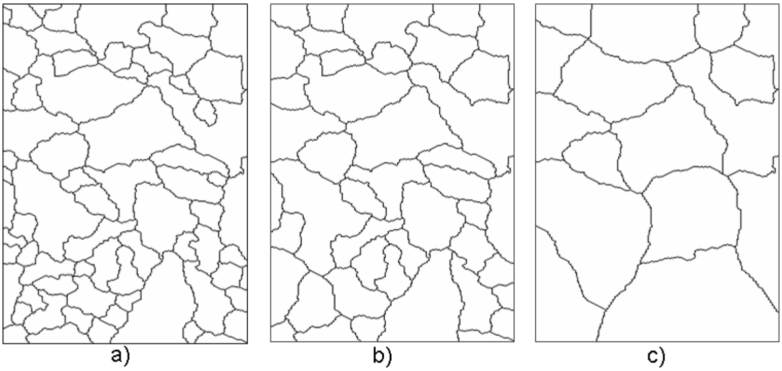


Fig. 7. a) Initial tessellation ψ, b) and c) thickening $\psi\gamma_{10}\psi\gamma_0$, and then $\psi\gamma_{20}\psi\gamma_{10}\psi\gamma_0$. The results are ordered according to both $\preccurlyeq$, and to the semi-group.

the seeds is preserved under growing. We must proceed by sequences of elementary operations which do maintain homotopy at each step (chap. XI-E in [10]) such as G. Bertrand's topological watersheds [2], in a binary and complemented version. Freedom is left for the succession of the elementary thickenings, so that one can well approximate the final equidistant cleavages of the Euclidean homologues. Moreover, Proposition 5 extends to $\mathcal{P}(\mathbb{Z}^2)$ when ψ is the opposite of a topological watershed.

3.2 Hierarchies of Thickenings Based on Connected Opening

Consider, in $\mathbb{R}^2$ or in $\mathbb{Z}^2$, a family $\{\gamma_j,\ j \in J\}$ of connected openings that depend on the integers $j \in J$, and the Voronoi thickening ψ. We now construct hierarchies of connected thickenings $\psi\gamma_j$. Remark firstly that $\subseteq$-decreasingness of the γ_i is equivalent to their $\preccurlyeq$-increasingness (the γ_i are connected openings, and each connected component of $\gamma_j(A)$ is also a connected component of $\gamma_i(A)$)

$$\{j \geq i \ \Rightarrow\ \gamma_j \subseteq \gamma_i\} \Leftrightarrow \{j \geq i \ \Rightarrow\ \gamma_j \succeq \gamma_i\}.$$

Hierarchies can be obtained in two ways, according as we focus on the increasingness of $j \to \gamma_j\psi$, or as we look for semi-groups. The second approach generates a more powerful structure, but requires sequences of operations.

Hierarchies of ordered operators

Proposition 6. *Let ψ be a thickening by zones of influence, and let $\{\gamma_j,\ j \in J\}$ be a $\preccurlyeq$-increasing family of connected openings, both in $\mathbb{R}^2$ or in $\mathbb{Z}^2$. Then the two thickenings $\{\psi\gamma_j,\ j \in J\}$ and $\{\gamma_j\psi,\ j \in J\}$ form two chains for the GBO:*

$$j \geq i \ \Rightarrow\ \psi\gamma_j \succcurlyeq \psi\gamma_i \ \ \textit{and} \ \ \gamma_j\psi \succcurlyeq \gamma_i\psi \qquad i, j \in J.$$

Hierarchies by semi-groups Consider the $\preccurlyeq$-connected thickening $\psi\gamma$ and let $\gamma = \gamma_j$ decrease according to $j \in J$,

$$j \geq i \quad \Rightarrow \gamma_j \leq \gamma_i, \qquad i, j \in J \ .$$

The connected components $\psi\gamma_j$ are unchanged under γ_i, and by idempotence of ψ, we obtain

$$j \geq i \quad \Rightarrow \ (\psi\gamma_i)(\psi\gamma_j) = (\psi\gamma_j).$$

Consequently, the $\{\psi\gamma_j\}$ generate, by sequential composition, the Matheron semi-group M_j:

$$M_i M_j = M_j M_i = M_j = (\psi\gamma_j)...(\psi\gamma_2)(\psi\gamma_1), \tag{8}$$

where the M_j are increasing for the GBO, since

$$j > i \quad \Rightarrow M_j = (\psi\gamma_j)...(\psi\gamma_{i+1})M_i \succ M_i.$$

Figure 7 illustrates such a progression.

Saliency and Hierarchy. Unlike the hierarchies based on the refinement ordering, those on the GBO involve two saliencies for each edge, at least in the case of Voronoi reconstructions that we study here. A new edge appears either at the lowest level, or when a new grain is generated at level i. Then it does not change as long as the two grains it separates are still present in the hierarchy, and disappears permanently when one of these grains vanishes, at level $j > i$.

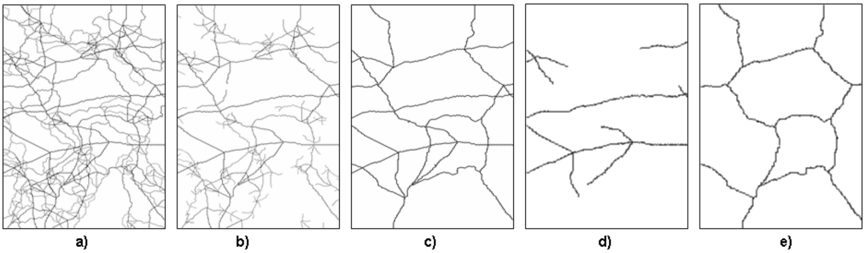


Fig. 8. a) and b), the two saliencies of a $\preceq$-hierarchy; c) and d), the respective cross sections of these saliencies at level 42; e) set diifference between c) and d), which results in partition 42 of the $\preceq$-hierarchy.

An example of this double saliency is depicted in Figure 8. A hierarchy has been produced by applying the semi group of operators (8) to the tessellation of Figure 7a). The pyramid is represented in a synthetic way by the two numerical functions of Figure 8a) and b). By selecting all frontiers darker than 42 in Figure 8a) we obtain the family of those frontiers that appear before step 42 (Fig. 8c)). Similarly, the threshold of 8b) at level 42 provides all frontiers that disappeared before step 42 (Fig. 8d)). The set difference between the two sections results in the partition at level 42 in the hierarchy (Fig. 8e)).

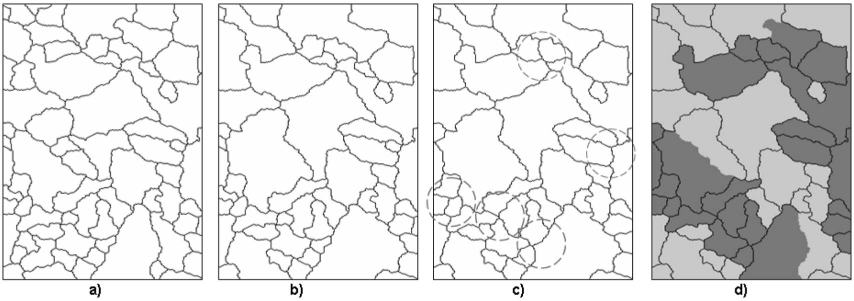


Fig. 9. a) partition D_i; b) partition $D_j \geq D_i$; c) $\psi\gamma(D_i)$, where the dotted rings indicate the places of changes; d) partition $D_j^* = D_j \sqcup \psi\gamma(D_i)$

Mixing two hierarchies. Segmentation processing often leads to hierarchies where a sequence of partitions is ordered by refinement (symbol $\leq$). This occurs, for example, in maps of watersheds when one weights the edges between adjacent basins according to their flooding level. Let $\{D_i, i \in I\}$ be such a sequence of partitions, with

$$i \leq j \quad \Rightarrow \quad D_i \leq D_j \quad \Rightarrow \quad D_i \preccurlyeq D_j$$

Consider a thickening $\psi\gamma$ that $\preccurlyeq$-enlarges D_i i.e. $D_i \preccurlyeq \psi\gamma(D_i)$. As $\psi\gamma$ is not $\preccurlyeq$-increasing, we cannot write $\psi\gamma(D_i) \preccurlyeq \psi\gamma(D_j)$; the hierarchical structure seems to be lost. However, the partition $\psi\gamma(D_i)$ is composed of the partial partition D_i' of all classes of D_i left unchanged under $\psi\gamma$, and of the partial partition D_i'' of all the other classes of $\psi\gamma(D_i)$. Let S' and S'' be the two corresponding supports, with $S' \cup S'' = E$. Take the restriction of D_j to set S' and that of $\psi\gamma(D_i)$ to set S'', and define by D_j^* the partition of E which is obtained by the concatenation $\sqcup$ of these two partial partitions:

$$D_j^* = (D_j)_{in\ S'} \ \sqcup (\psi\gamma(D_i))_{in\ S''}.$$

The partition D_j^* is equal to $\psi\gamma(D_i)$ in S'', and $\preccurlyeq$-larger than $\psi\gamma(D_i)$ elsewhere, hence $D_j^* \succeq \psi\gamma(D_i)$. Moreover, D_j^* is invariant under $\psi\gamma$, since all its classes are invariant under γ. We can write

$$D_i \preccurlyeq \psi\gamma(D_i) \preccurlyeq D_j^* = \psi\gamma(D_j^*).$$

Suppose now that the γ's are themselves ordered, i.e. that they form the granulometry $\{\gamma_p, p \in P\}$. Then, for $p \leq q$ we can write

$$p \leq q \text{ and } \ i \leq j \quad \Rightarrow \quad D_i \preccurlyeq \psi\gamma_p(D_i) \preccurlyeq \psi\gamma_p(D_j^*) \preccurlyeq \psi\gamma_q\psi\gamma_p(D_j^*).$$

We find again the semi-goup (8).

$\preccurlyeq$-thickening a low level. The GBO can also serve as a tool for filtering. Consider a level D_{i_0} in the hierarchy $\{D_i, i \in I\}$ that we want to $\preccurlyeq$-amend for reducing

its small particles (e.g. Figure 10b)). One can perform some $\preccurlyeq$-thickening $\psi\gamma$, which produces the new partition $\psi\gamma(D_{i_0})$ of Figure 10c), and apply to $\psi\gamma(D_{i_0})$ the criterion which already allowed us to suppress edges in the initial hierarchy $\{D_i, i \in I\}$. Indeed, one can check by comparing Figures 10 a) and c) that most of the long previous edges are still in place, in the case of this example at least.a) Initial tessellation; b) additional noisy small classes; c) thickening $\psi\mu$ of b), by a dodecagonal opening γ of size 5.

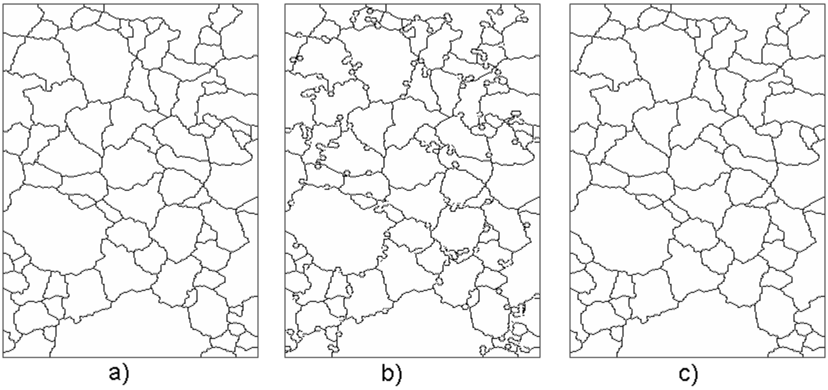


Fig. 10. a) Initial tessellation; b) additional noisy small classes; c) thickening $\psi\mu$ of b), by a dodecagonal opening γ of size 5

4 Conclusion

The grain building ordering presented here, as well as the other orderings studied [8] model how partitions of the space are reorganized, and enlarged, in some physical processes. It does it in a more realistic way than the usual refinement ordering, but in compensation, it leads to less simple properties (e.g. two saliencies instead of one). In practice, it allows to eliminate small parasite classes in partitions, and also to "average" different partitions closed enough to each other (this last point, not presented above, was already developed in [12]).

Acknowledgement

The author wishes to thank *J. Angulo, J. Cousty, L. Najman, and Ch. Ronse* for their valuable comments.

References

1. Angulo, J., Serra, J.: Modeling and segmentation of colour images in polar representations. IVC 25, 475–495 (2007)
2. Bertand, G.: On topological watersheds. JMIV 22(2-3), 217–230 (2005)

3. Crespo, J., Schafer, R.W., Serra, J., Gratin, C., Meyer, F.: The flat zone approach: A general low-level region merging segmentation method. Sig. Proc. 62(1), 37–60 (1997)
4. Lantuejoul, C.: Skeletonization in quantitative metallography. In: Haralick, R.M., Simons, J.-C. (eds.) Issues of Digital Image Processing, Sijthoff and Noordhoff (1980)
5. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. IEEE Trans. on PAMI 18(12), 1163–1173 (1996)
6. Noyel, G., Angulo, J., Jeulin, D.: On distances, paths and connections for hyperspectral image segmentation. In: Proceedings of the 8th ISMM. MCT/INPE, vol. 1, pp. 399–410 (2007)
7. Ronse, C.: Partial partitions, partial connections and connective segmentation. JMIV 32, 97–125 (2008)
8. Ronse, C.: Orders on partial partitions and maximal partitioning of sets. In: ISMM 2011 (2011)
9. Salembier, P., Garrido, L., Garcia, D.: Auto-dual connected operators based on iterative merging algorithms. In: Heijmans, H., Roerdink, J. (eds.) ISMM 1998, pp. 183–190. Kluwer, Dordrecht (1998)
10. Serra, J.: Image Analysis and Mathematical Morphology. Ac. Press, London (1982)
11. Serra, J.: A lattice approach to Image segmentation. JMIV 24, 83–130 (2006)
12. Serra, J.: Morphological Operators for the Segmentation of Colour Images. In: Bilodeau, M., Meyer, F., Schmitt, M. (eds.) Space, Structure, and Randomness. Lecture Notes in Statistics, vol. 183, pp. 223–255. Springer, Heidelberg (2005)
13. Serra, J.: Ordre de la construction et segmentations hiérarchiques, Colloque ESIEE (April 2, 2010), http://laurentnajman.org/serra70/index
14. Soille, P., Grazzini, J.: Constrained Connectivity and Transition Regions. In: Wilkinson, M.H.F., Roerdink, B.T.M. (eds.) ISMM 2009, pp. 59–70. Springer, Heidelberg (2009)

Orders on Partial Partitions and Maximal Partitioning of Sets[⋆]

Christian Ronse

LSIIT UMR 7005 CNRS-UdS
Parc d'Innovation, Boulevard Sébastien Brant
BP 10413, 67412 ILLKIRCH CEDEX, France
cronse@unistra.fr
http://lsiit-cnrs.unistra.fr/miv

Abstract. The segmentation of a function on a set can be considered as the construction of a maximal partial partition of that set with blocks satisfying some criterion for the function. Several order relations on partial partitions are considered in association with types of operators and criteria involved in the segmentation process. We investigate orders for which this maximality of the segmentation partial partition is preserved in compound segmentation with two successive criteria. Finally we consider valuations on partial partitions, that is, strictly isotone functions with positive real values; this gives an alternative approach where the valuation, not the partial partition, should be maximized.

1 Introduction

The purpose of this communication is to discuss order-theoretic issues involved in image segmentation. We consider images as functions $E \to T$, where E is the space of points and T is the set of image values. A *partition* of E is a family of nonvoid mutually disjoint subsets of E, called *blocks*, whose union is E. Soille [15] summarizes conventional requirements of image segmentation as follows:

1. The segmentation method relies on a criterion that determines, for every function F and every subset A of E, whether F is homogeneous on A or not.
2. Given a function F, its segmentation is a partition of E into connected blocks on which F is homogeneous; these blocks are called segmentation classes.
3. Merging two or more adjacent segmentation classes, F is not homogeneous on the resulting set; in other words F cannot be homogeneous on a connected union of two or more segmentation classes.

Let us formalize these principles. We first recall some terminology. Write $\Pi(E)$ for the set of all partitions of E. Now $\Pi(E)$ is ordered by *refinement*: for $\pi_1, \pi_2 \in \Pi(E)$, we say that π_1 is *finer* than π_2, or that π_2 is *coarser* than π_1, and write $\pi_1 \leq \pi_2$ (or $\pi_2 \geq \pi_1$), iff every block of π_1 is included in a block of π_2, that

[⋆] This work received funding from the Agence Nationale de la Recherche, contract ANR-2010-BLAN-0205-01.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 49–60, 2011.

is, every block of π_2 is a union of blocks of π_1. Then $(\mathit{\Pi}(E), \leq)$ is a complete lattice. For any family $\mathcal{C} \subseteq \mathcal{P}(E)$, let $\mathit{\Pi}(E, \mathcal{C}) = \mathit{\Pi}(E) \cap \mathcal{P}\big(\mathcal{C} \setminus \{\emptyset\}\big)$ be the family of all partitions whose blocks belong to $\mathcal{C}$ (in fact, to $\mathcal{C} \setminus \{\emptyset\}$).

The connectivity of sets is given by a *connection* $\mathcal{C}$ on $\mathcal{P}(E)$ [10,4,9]. By item 1, the segmentation method is based on a *criterion* [12,9], that is a map $\mathsf{cr} : T^E \times \mathcal{P}(E) \to \{0, 1\}$, where for $F : E \to T$ and $A \in \mathcal{P}(E)$, we have $\mathsf{cr}[F, A] = 1$ if A is connected (i.e., $A \in \mathcal{C}$) *and* F is homogeneous on A. For any $F : E \to T$, we obtain the family $\mathcal{C}^F_{\mathsf{cr}}$ of all connected sets on which F is homogeneous, that is,

$$\mathcal{C}^F_{\mathsf{cr}} = \{A \in \mathcal{P}(E) \mid \mathsf{cr}[F, A] = 1\} \ . \tag{1}$$

By item 2, the segmentation of F is a partition π^F_{cr} of E with blocks in $\mathcal{C}^F_{\mathsf{cr}}$, that is, $\pi^F_{\mathsf{cr}} \in \mathit{\Pi}(E, \mathcal{C}^F_{\mathsf{cr}})$, and by item 3, for any partition π strictly coarser than π^F_{cr}, the blocks of π cannot belong to $\mathcal{C}^F_{\mathsf{cr}}$, that is, $\pi^F_{\mathsf{cr}} < \pi \ \Rightarrow \ \pi \notin \mathit{\Pi}(E, \mathcal{C}^F_{\mathsf{cr}})$. Thus the segmentation of F is a maximal element of $\mathit{\Pi}(E, \mathcal{C}^F_{\mathsf{cr}})$ for the refinement order.

As remarked by Serra [12,9], it is necessary to consider the segmentation of a function F not only on E, but on any subset A of E. Thus in the above conditions, we have to consider the segmentation of a function F on a subset A of E, which is a partition of A. Here the segmentation method based on criterion cr associates to every function $F : E \to T$ and subset A of E the segmentation $\sigma^F_{\mathsf{cr}}(A)$, which is a partition of A; we require then that $\sigma^F_{\mathsf{cr}}(A)$ is a maximal element of $\mathit{\Pi}(A, \mathcal{C}^F_{\mathsf{cr}})$ for the refinement ordering. We call this requirement the *maximality principle*.

This wider approach has the advantage of allowing to consider the segmentation process as an operator acting on the lattice of partitions [7]. Indeed, we have introduced here a *set splitting operator* σ^F_{cr} that maps each $A \in \mathcal{P}(E)$ on a partition of A. From σ^F_{cr} we derive the *block splitting operator* $\beta(\sigma^F_{\mathsf{cr}}) : \mathit{\Pi}(E) \to \mathit{\Pi}(E)$ that acts on a partition π by applying σ^F_{cr} to each block of π, in other words $\beta(\sigma^F_{\mathsf{cr}})(\pi) = \bigcup_{B \in \pi} \sigma^F_{\mathsf{cr}}(B)$, see [7]. Then the maximality principle means that $\beta(\sigma^F_{\mathsf{cr}})(\pi)$ is a maximal element of the set of all $\pi' \in \mathit{\Pi}(E, \mathcal{C}^F_{\mathsf{cr}})$ such that $\pi' \leq \pi$.

Now this order-theoretic approach to image segmentation has been generalized to partial partitions [5,9]. A *partial partition* of E is a partition of any subset of E; in other words, it is a family of nonvoid mutually disjoint subsets of E, called *blocks*, but here we do no more assume that their union covers E. In parallel, the theory of connections has been generalized to that of *partial connections* [5]. Indeed, apart from the obvious fact that some segmentation algorithms produce a partial partition rather than a partition (the points of E not covered by it constitute the borders between regions), the "partial" framework is more versatile, allowing to represent individual or multiple set markers, as well as the progressive steps in the construction of a segmentation; also the construction of new connections [5] or new operators on partitions [6] is easier with the use of partial connections and partial partitions.

Write $\mathit{\Pi}^*(E)$ for the set of all partial partitions of E. Write $\boldsymbol{\varnothing}$ for the empty partial partition (with no block), and for any $A \in \mathcal{P}(E) \setminus \{\emptyset\}$, let $\mathbf{1}_A = \{A\}$ and $\mathbf{0}_A = \big\{\{p\} \mid p \in A\big\}$, while $\mathbf{1}_\emptyset = \mathbf{0}_\emptyset = \boldsymbol{\varnothing}$ [5]. For $\pi \in \mathit{\Pi}^*(E)$, the *support* of π, written $\mathsf{supp}(\pi)$, is the union of its blocks: $\mathsf{supp}(\pi) = \bigcup \pi$; the complement $E \setminus \mathsf{supp}(\pi)$ of the support is the *background* of π. For $\mathcal{C} \subseteq \mathcal{P}(E)$, let $\Pi^*(E,\mathcal{C}) = \Pi^*(E) \cap \mathcal{P}\big(\mathcal{C} \setminus \{\emptyset\}\big)$. The refinement order on $\Pi(E)$ extends naturally to $\Pi^*(E)$: for $\pi_1, \pi_2 \in \Pi^*(E)$, we write $\pi_1 \leq \pi_2$ (or $\pi_2 \geq \pi_1$), iff every block of π_1 is included in a block of π_2. Now contrarily to $\Pi(E)$, for $\pi_2 \geq \pi_1$ a block of π_2 is not necessarily a union of blocks of π_1, it can also contain points outside the support of π_1, it can even contain no block of π_1; thus a "coarsening" of a partial partition results not only from merging blocks, but also from inflating individual blocks or creating new blocks. Hence this order relation on $\Pi^*(E)$ should not be called refinement, a more appropriate denomination could be *extended refinement*; we will simply call it the *standard order*. Its main advantages are that (1) it naturally constitutes $\Pi^*(E)$ into a complete lattice, (2) its restriction to $\Pi(E)$ coincides with the refinement order, and the non-empty supremum and infimum operations in $\Pi(E)$ are inherited from those in $\Pi^*(E)$, and (3) we easily obtain a nice theory about idempotent block splitting operators [7,8].

Let us recall here that each order relation R is identified with the set of ordered pairs (a, b) such that $a\ R\ b$, so if we say that the order S is included in the order R, or that R contains S, this means that $a\ S\ b \ \Rightarrow\ a\ R\ b$.

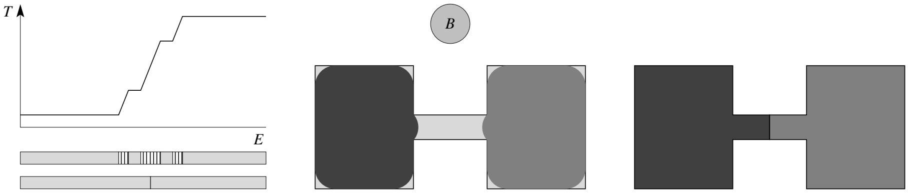


Fig. 1. Left: the graph of a one-dimensional grey-level edge; below we show its segmentation into connected classes with bounded slope (light grey rectangles for non-singleton classes, vertically hatched ones for groups of singleton classes); notice the large number of small classes on the edge; eliminating them, the final segmentation (bottom) consists of the influence zones of the two large classes. Middle: a subset of the plane is segmented into two connected zones open by a disk B, while the remaining points form singletons; right: the desired segmentation is obtained by the the influence zones of the two open zones.

Serra [13,14] noticed that in many image segmentation algorithms, "small parasitic" segmentation classes appear along contours and transitions, where the region homogeneity criterion fails; they can be eliminated and then one can take as final partition the influence zones of the significant segmentation classes corresponding to objects. See Figure 1. In order to analyse this process, he defined the *building order* $\Subset$ on $\Pi^*(E)$ as follows[1]: $\pi_1 \Subset \pi_2$ iff every block of π_2 contains at least one block of π_1. Then $\Subset$ is a partial order relation, and it is generally unrelated to the standard order $\leq$, except when the partial partitions

[1] In fact, Serra wrote $\pi_1 \preceq \pi_2$, but we will use the symbol $\prec$ for the covering relation.

have the same support: if $\pi_1 \leq \pi_2$ and $\mathsf{supp}(\pi_1) = \mathsf{supp}(\pi_2)$, then $\pi_1 \Subset \pi_2$; in particular for partitions, the building order $\Subset$ contains the refinement order $\leq$. However the building order does not constitute a lattice, and it is not easy to define operators with given order-theoretic properties (for example, isotony). Serra constructs extensive operators for the building order in two steps; starting from a partial partition π_0:

1. Remove "small parasitic" blocks from π_0 (through some "parasitism" and size criterion); the resulting partial partition π_1 satisfies $\pi_0 \supseteq \pi_1$, thus $\pi_0 \geq \pi_1$, but $\pi_0 \Subset \pi_1$.
2. Inflate the blocks of π_1 (for example by a SKIZ), without creating any new block; the merging of blocks is not excluded, but it is not used in practice; the resulting partial partition π_2 satisfies both $\pi_1 \leq \pi_2$ and $\pi_1 \Subset \pi_2$.

Then the partial partition π_2, having fewer but bigger blocks than π_0, is "better", a quality that is certified by the order $\pi_0 \Subset \pi_2$.

We remark that the construction of π_2 from π_0 involves two operations using two distinct criteria, and two distinct orders included in $\Subset$, which are at the same time included in $\geq$ and $\leq$ respectively. Furthermore, although $\pi_0 \Subset \pi_1$, π_1 is not considered as a "good" result; in all practical examples, the block growth of step 2 must be repeated until the blocks removed in step 1 are fully covered, in other words $\mathsf{supp}(\pi_2) = \sup(\pi_0)$ (in fact, Serra considers that π_0 and π_2 are partitions of E).

We propose that a meaningful order relation on partial partitions should be viewed through the family of operations that "enlarge" a partial partition, and these operations should effectively be involved in image segmentation techniques; if possible, they should be linked with segmentation criteria. We will indeed obtain several relevant orders included in the standard order $\leq$. Following the *maximality principle*, namely that the segmentation of $F : E \to T$ on $A \subseteq E$ following a criterion cr, is a maximal element of $\Pi^*(A, \mathcal{C}^F_{\mathsf{cr}})$, we show that for some of these orders, this principle is preserved in the compound segmentation paradigm [11,9], where after a first segmentation, a second one (with another criterion) is applied to the residue. Finally we investigate *valuations* on partial partitions, that is, strictly isotone maps $\Pi^*(E) \to \mathbb{R}^+$, and the possibility to replace "maximal" by "having maximal valuation" in the maximality principle.

2 Partial Order Relations on $\Pi^*(E)$

We will investigate several partial order relations on $\Pi^*(E)$. Each order is denoted by a variant of $\leq$, such as $\underline{\blacktriangleleft}$, with the associated notation $\blacktriangleleft$ for the corresponding strict order, and the mirror notation $\underline{\blacktriangleright}$ for the inverse order and $\blacktriangleright$ for the inverse strict order. We first propose three *primary* partial order relations, from which other ones will be built:

1. The *inclusion* order $\subseteq$: $\pi_1 \subseteq \pi_2$ iff every block of π_1 is a block of π_2. This order is involved in the elimination of "parasitic" segmentation classes, cf.

Serra's step 1 above, but also in the compound segmentation paradigm, where we add to the blocks of a first segmentation those of a second segmentation of the residue.

2. The *pure refinement* order $\sqsubseteq$: $\pi_1 \sqsubseteq \pi_2$ iff $\pi_1 \leq \pi_2$ and $\mathsf{supp}(\pi_1) = \mathsf{supp}(\pi_2)$; in other words, every block of π_1 is included in a block of π_2, and every block of π_2 is a union of blocks of π_1; we say then that π_1 is *purely finer* than π_2, or that π_2 is *purely coarser* than π_1. This order is involved in split-and-merge operations in segmentation.
3. The *inflating* order $\trianglelefteq$: $\pi_1 \trianglelefteq \pi_2$ iff $\pi_1 \leq \pi_2$ and every block of π_2 contains exactly one block of π_1, in other words the inclusion relation between blocks of π_1 and those of π_2 is a bijection; we say then that π_1 is a *deflation* of π_2, or that π_2 is an *inflation* of π_1. This order is involved in parallel region growing (SKIZ, watershed), cf. Serra's step 2 above.

Next we give two *secondary* partial order relations; as their names (and notation) suggest, they can be constructed by composing two of the three primary orders, see Proposition 1:

4. The *inclusion-inflating* order $\mathrel{\underline{\subset\!\!\!\lhd}}$ is defined by $\pi_1 \mathrel{\underline{\subset\!\!\!\lhd}} \pi_2$ iff $\pi_1 \leq \pi_2$ and every block of π_2 contains at most one block of π_1.
5. The *refinement-inflating* order $\mathrel{\underline{\sqsubset\!\!\!\lhd}}$ is the intersection of the building and standard orders, it is defined by $\pi_1 \mathrel{\underline{\sqsubset\!\!\!\lhd}} \pi_2$ iff $\pi_1 \leq \pi_2$ and every block of π_2 contains at least one block of π_1.

The above five order relations are included in the standard order $\leq$, in other words, each such relation on (π_1, π_2) implies that $\pi_1 \leq \pi_2$. More precisely:

Proposition 1. *Here π_1, π_2, π, π' designate arbitrary partial partitions of E.*

1. *The standard order contains the inclusion, pure refinement and inflating orders: each of $\pi_1 \subseteq \pi_2$, $\pi_1 \sqsubseteq \pi_2$ and $\pi_1 \trianglelefteq \pi_2$ implies $\pi_1 \leq \pi_2$. It is generated by inclusion followed by pure refinement: $\pi_1 \leq \pi_2 \iff \exists\, \pi,\ \pi_1 \subseteq \pi \sqsubseteq \pi_2$.*
2. *The inclusion-inflating order contains the inclusion and inflating orders: each of $\pi_1 \subseteq \pi_2$ and $\pi_1 \trianglelefteq \pi_2$ implies $\pi_1 \mathrel{\underline{\subset\!\!\!\lhd}} \pi_2$. It it is generated by composing them in any order:*

$$\pi_1 \mathrel{\underline{\subset\!\!\!\lhd}} \pi_2 \iff \big(\exists\, \pi,\ \pi_1 \subseteq \pi \trianglelefteq \pi_2\big) \iff \big(\exists\, \pi',\ \pi_1 \trianglelefteq \pi' \subseteq \pi_2\big) \ .$$

3. *The refinement-inflating order contains the pure refinement and inflating orders: each of $\pi_1 \sqsubseteq \pi_2$ and $\pi_1 \trianglelefteq \pi_2$ implies $\pi_1 \mathrel{\underline{\sqsubset\!\!\!\lhd}} \pi_2$. It is generated by composing them in any order:*

$$\pi_1 \mathrel{\underline{\sqsubset\!\!\!\lhd}} \pi_2 \iff \big(\exists\, \pi,\ \pi_1 \sqsubseteq \pi \trianglelefteq \pi_2\big) \iff \big(\exists\, \pi',\ \pi_1 \trianglelefteq \pi' \sqsubseteq \pi_2\big) \ .$$

4. *The building order contains the inverse inclusion, pure refinement and inflating orders: each of $\pi_1 \supseteq \pi_2$, $\pi_1 \sqsubseteq \pi_2$ and $\pi_1 \trianglelefteq \pi_2$ implies $\pi_1 \Subset \pi_2$. It is generated by inverse inclusion followed by inflating: $\pi_1 \Subset \pi_2 \iff \exists\, \pi,\ \pi_1 \supseteq \pi \trianglelefteq \pi_2$.*

Each primary order on $\Pi^*(E)$ corresponds to the operations on blocks used for enlarging a partial partition; thus the inclusion $\subseteq$, inverse inclusion $\supseteq$, pure refinement $\sqsubseteq$ and inflating $\trianglelefteq$ orders correspond respectively to: creating, removing, merging and inflating blocks. We show in Figure 2 the sequence of two such operations involved in the orders $\leq$, $\underline{\subset\!\!\lhd}$, $\underline{\sqsubset\!\!\lhd}$ and $\Subset$.

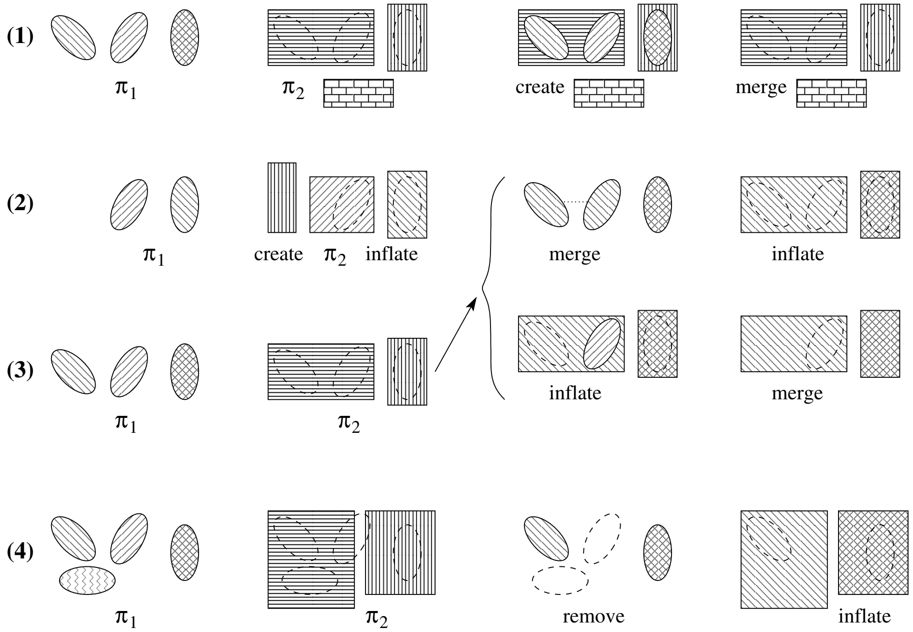


Fig. 2. In each partial partition, blocks are distinguished by their hatching. (1) $\pi_1 \leq \pi_2$; π_2 is obtained from π_1 by creating, then merging blocks. (2) $\pi_1 \underline{\subset\!\!\lhd} \pi_2$; π_2 is obtained from π_1 by creating blocks outside $\mathsf{supp}(\pi_1)$ and at the same time inflating blocks of π_1. (3) $\pi_1 \underline{\sqsubset\!\!\lhd} \pi_2$; π_2 is obtained from π_1 by merging, then inflating blocks, or vice versa. (4) $\pi_1 \Subset \pi_2$; π_2 is obtained from π_1 by removing, then inflating blocks.

In watershed segmentation, the *saliency* of a watershed line between two basins is the flooding level at which these two basins merge [2]. More generally, given a *hierarchy of segmentations*, that is, an increasing sequence $\mathbf{0}_E = \pi_0 \leq \ldots \leq \pi_n = \mathbf{1}_E$ of segmentation partitions (with connected blocks), given two neighbouring blocks $B, C \in \bigcup_{i=0}^{n} \pi_i$, the *saliency* of the edge separating B and C is the level i at which this edge disappears, in other words the least level i such that B and C are included in the same block of π_i [1]. For partial partitions, things are more complicated, because the growth of the supports can lead to the creation of new edges. Between two neighbouring digital points p, q lies an edge element $e(p, q)$ (line in 2D, surface in 3D). In a partial partition π, any edge element belongs to one of the following 4 categories:

1. *background*: lying between two points of the background of π;
2. *outer*: lying between a point of the support and one of the background of π;
3. *separating*: lying between two points belonging to two distinct blocks of π;
4. *inner*: lying between two points belonging to the same block of π.

In a hierarchy $\mathbf{\emptyset} = \pi_0 \leq \ldots \leq \pi_n = \mathbf{1}_E$ of segmentation partial partitions (for the standard order), as the growth adds points to the support and merges blocks, each edge element can change category only in increasing order: *background* < *outer* < *separating* < *inner*. Thus the saliency associated to an edge element on a block boundary is given by two numbers, the level where it appears (transition *background* → *outer* or *separating*), and the higher level where it disappears (transition *outer* or *separating* → *inner*); for edge elements not belonging to block boundaries, we have the transition *background* → *inner*.

Alternately, the saliency $s(p)$ of a point $p \in E$ is the least level i such that $p \in \mathsf{supp}(\pi_i)$, and the saliency $s(p,q)$ of an edge element $e(p,q)$ is the least level j such that p and q belong to the same block of π_j. For two neighbouring pixels p,q, the 3 salencies $s(p)$, $s(q)$ and $s(p,q)$ satisfy $s(p,q) \geq \max(s(p), s(q))$, and they determine for each level the category of $e(p,q)$: for $i < \min(s(p), s(q))$: *background*; for $\min(s(p), s(q)) \leq i < \max(s(p), s(q))$: *outer*; for $\max(s(p), s(q)) \leq i < s(p,q)$: *separating*; for $s(p,q) \leq i$: *inner*.

3 Maximality in Compound Segmentation

A *set splitting operator on* $\mathcal{P}(E)$ is a map $\sigma : \mathcal{P}(E) \to \mathit{\Pi}^*(E)$ such that for every $X \in \mathcal{P}(E)$, $\sigma(X) \in \mathit{\Pi}^*(X)$; the corresponding *block splitting operator on* $\mathit{\Pi}^*(E)$ is the map $\beta(\sigma) : \mathit{\Pi}^*(E) \to \mathit{\Pi}^*(E) : \pi \mapsto \bigcup_{B \in \pi} \sigma(B)$ (i.e., it applies σ to each block of a partial partition) [7]. Given a partial order relation $\underline{\blacktriangleleft}$ on $\mathit{\Pi}^*(E)$, a family $\mathcal{C} \subseteq \mathcal{P}(E)$, and a set splitting operator σ on $\mathcal{P}(E)$, we say that σ *is* $\mathcal{C}$*-maximal for* $\underline{\blacktriangleleft}$ if for every $A \in \mathcal{P}(E)$, $\sigma(A)$ is a maximal element of $\mathit{\Pi}^*(A, \mathcal{C})$ according to the order $\underline{\blacktriangleleft}$.

For a segmentation method based on a criterion cr, to any function $F : E \to T$ is associated the family $\mathcal{C}^F_{\mathsf{cr}}$ of "homogeneous sets", see (1), and the set splitting operator σ^F_{cr}, where for $A \in \mathcal{P}(E)$, $\sigma^F_{\mathsf{cr}}(A)$ is the segmentation of F on A. Now the maximality principle states that for every function $F : E \to T$, σ^F_{cr} is $\mathcal{C}^F_{\mathsf{cr}}$-maximal for the chosen order (usually the standard order $\leq$ [7], but possibly the building order $\Subset$ [13]).

In *connective segmentation* [12,5,9], each $\mathcal{C}^F_{\mathsf{cr}}$ is a partial connection, equivalently, for every $A \in \mathcal{P}(E)$, $\mathit{\Pi}^*(A, \mathcal{C}^F_{\mathsf{cr}})$ has a greatest element (for the standard order $\leq$), namely the partial partion of A into its $\mathcal{C}^F_{\mathsf{cr}}$-components. Thus by the maximality principle, the segmentation $\sigma^F_{\mathsf{cr}}(A)$ must necessarily be that greatest element. In fact, this corresponds to the case where $\beta(\sigma^F_{\mathsf{cr}})$ is an opening [7]. Since partial connections constitute a complete lattice for the inclusion order, such connective segmentations also form a complete lattice [12,5,9]; for a family cr_i $(i \in I)$ of connective criteria, the infimum of the connective segmentation operators $\sigma^F_{\mathsf{cr}_i}$ is $\sigma^F_{\inf_{i \in I} \mathsf{cr}_i}$, and their supremum is $\bigvee_{i \in I} \sigma^F_{\mathsf{cr}_i}$.

There is a priori no such lattice structure in the non-connective case. However the maximality principle is consistent with Serra's compound segmentation paradigm [11,9], where after a first segmentation producing a partial partition, the residue (the set of points not covered by that partial partition) is partitioned by another segmentation method, and the union of the two partial partitions forms the compound segmentation. This operation has been formalized as follows [8]. Let σ_1, σ_2 be two set splitting operators on $\mathcal{P}(E)$; the *residual combination of* σ_1 *followed by* σ_2 is the set splitting operator $\sigma_1 \ltimes \sigma_2$ on $\mathcal{P}(E)$ defined by $(\sigma_1 \ltimes \sigma_2)(X) = \sigma_1(X) \cup \sigma_2\big(X \setminus \mathsf{supp}[\sigma_1(X)]\big)$ for any $X \in \mathcal{P}(E)$. Note that in [8] we wrote it $\mathsf{rc}[\sigma_1, \sigma_2]$.

Let us define the binary operation $\leftthreetimes$ on subsets of $\mathcal{P}(E)$ by setting for any $\mathcal{H}_1, \mathcal{H}_2 \subseteq \mathcal{P}(E)$:

$$\mathcal{H}_1 \leftthreetimes \mathcal{H}_2 = \mathcal{H}_1 \cup \big\{X \in \mathcal{H}_2 \mid \forall Y \in \mathcal{P}(X) \setminus \{\emptyset\},\ Y \notin \mathcal{H}_1\big\} \ .$$

In other words $\mathcal{H}_1 \leftthreetimes \mathcal{H}_2 = \mathcal{H}_1 \cup \big\{X \in \mathcal{H}_2 \mid \mathcal{P}(X) \cap \mathcal{H}_1 \subseteq \{\emptyset\}\big\}$. Let us say that a partial order relation $\underline{\blacktriangleleft}$ on $\Pi^*(E)$ is *well-composed* if for any $\pi_0, \pi_1, \pi_2 \in \Pi^*(E)$ we have:

$$\pi_0 \subseteq \pi_1 \implies \pi_0 \underline{\blacktriangleleft} \pi_1 \implies \pi_0 \leq \pi_1 ,$$
$$\big(\pi_0 \leq \pi_1 \subseteq \pi_2 \text{ and } \pi_0 \underline{\blacktriangleleft} \pi_2\big) \implies \pi_0 \underline{\blacktriangleleft} \pi_1 \ .$$

For example the inclusion $\subseteq$, inclusion-inflating $\trianglelefteq$ and standard $\leq$ orders are well-composed.

Theorem 2. *Let $\underline{\blacktriangleleft}$ be a well-composed partial order relation on $\Pi^*(E)$, let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{P}(E)$, and let σ_1, σ_2 be set splitting operators on $\mathcal{P}(E)$, respectively $\mathcal{C}_1$-maximal and $\mathcal{C}_2$-maximal for $\underline{\blacktriangleleft}$. Then $\sigma_1 \ltimes \sigma_2$ is $\mathcal{C}_1 \leftthreetimes \mathcal{C}_2$-maximal for $\underline{\blacktriangleleft}$.*

The residual combination $\ltimes$ is associative [8], thus we can define $\sigma_1 \ltimes \cdots \ltimes \sigma_n$ for any $n \geq 2$, and we have for any $X \in \mathcal{P}(E)$:

$$(\sigma_1 \ltimes \cdots \ltimes \sigma_n)(X) = \bigcup_{i=1}^{n} \sigma_i(X_i), \quad \text{where}$$
$$X_1 = X \quad \text{and} \quad \text{for } i = 2, \ldots, n,\ X_i = X \setminus \bigcup_{j=1}^{i-1} \mathsf{supp}\big[\sigma_j(X_j)\big] \ .$$

We can easily show that the operation $\leftthreetimes$ is also associative, and we have

$$\mathcal{H}_1 \leftthreetimes \cdots \leftthreetimes \mathcal{H}_n = \mathcal{H}'_1 \cup \cdots \cup \mathcal{H}'_n, \quad \text{where} \quad \mathcal{H}'_1 = \mathcal{H}_1 \quad \text{and}$$
$$\text{for } i = 2, \ldots, n,\ \mathcal{H}'_i = \Big\{X \in \mathcal{H}_i \mid \forall Y \in \mathcal{P}(X) \setminus \{\emptyset\},\ Y \notin \bigcup_{j=1}^{i-1} \mathcal{H}_j\Big\} \ .$$

Corollary 3. *Let $\underline{\blacktriangleleft}$ be a well-composed partial order relation on $\Pi^*(E)$, let $\mathcal{C}_1, \ldots, \mathcal{C}_n \subseteq \mathcal{P}(E)$ ($n \geq 2$), and let $\sigma_1, \ldots, \sigma_n$ be set splitting operators on $\mathcal{P}(E)$ such that for each $i = 1, \ldots, n$, σ_i is $\mathcal{C}_i$-maximal for $\underline{\blacktriangleleft}$. Then $\sigma_1 \ltimes \cdots \ltimes \sigma_n$ is $\mathcal{C}_1 \leftthreetimes \cdots \leftthreetimes \mathcal{C}_n$-maximal for $\underline{\blacktriangleleft}$.*

4 Block Splitting Thinnings

We will consider now the possible idempotence of $\beta(\sigma)$ for a $\mathcal{C}$-maximal set splitting operator σ on $\mathcal{P}(E)$. But let us first briefly introduce some general terminology. Consider a poset $(P, \leq)$; for any operator π on P, let $\mathsf{Inv}(\psi) = \{x \in P \mid \psi(x) = x\}$ be the *invariance domain* of ψ. A *thinning* is an anti-extensive idempotent operator. A *max-thinning* is an operator ψ such that for all $x \in P$, $\psi(x)$ is a maximal element of $\{y \in \mathsf{Inv}(\psi) \mid y \leq x\}$. A *C-thinning* [8] (in [3] we said an *open-condensation*) is an anti-extensive operator ψ such that for any $x, y \in P$, $\big[\psi(x) \leq y \leq x\big] \;\Rightarrow\; \psi(y) = \psi(x)$. Then every C-thinning is a max-thinning, and every max-thinning is a thinning.

Proposition 4. *Let $\trianglelefteq$ be a well-composed partial order relation on $\Pi^*(E)$ and let $\mathcal{C} \subseteq \mathcal{P}(E)$. A set splitting operator σ on $\mathcal{P}(E)$ is $\mathcal{C}$-maximal for $\trianglelefteq$ iff for any $\pi \in \Pi^*(E)$, $\beta(\sigma)(\pi)$ is a maximal element (for $\trianglelefteq$) of $\{\pi' \in \Pi^*(E, \mathcal{C}) \mid \pi' \leq \pi\}$.*

In fact, $\beta(\sigma)$ will be idempotent iff for every $A \in \mathcal{C} \setminus \{\emptyset\}$, $\mathbf{1}_A$ is the greatest element (for $\trianglelefteq$) of $\Pi^*(A, \mathcal{C})$.

Let us now consider the particular case where $\trianglelefteq$ is the standard order $\leq$. Following Proposition 4, we see that a set splitting operator σ on $\mathcal{P}(E)$ is $\mathcal{C}$-maximal for $\leq$ iff $\beta(\sigma)$ is a max-thinning with $\mathsf{Inv}(\beta(\sigma)) = \Pi^*(E, \mathcal{C})$.

In [8] we showed that given two set splitting operators σ_1, σ_2 such that $\beta(\sigma_1)$ and $\beta(\sigma_2)$ are C-thinnings on $\Pi^*(E, \mathcal{C})$, then $\beta(\sigma_1 \ltimes \sigma_2)$ will be a C-thinning on $\Pi^*(E, \mathcal{C})$. This result was in fact a particular case of a general construction on posets, of which another special case was a result of [3]: given two C-thinnings θ_1, θ_2 on $\mathcal{P}(E)$, the operator on $\mathcal{P}(E) : X \mapsto \theta_1(X) \cup \theta_2\big(X \setminus \theta_1(X)\big)$ will be a C-thinning. However Theorem 2 is specific to $\Pi^*(E)$, the analogous result on $\mathcal{P}(E)$ is not valid. For example, let $E = \mathbb{Z}^2$ and define the operator θ on $\mathcal{P}(E)$ that extracts from any $X \in \mathcal{P}(E)$ a rectangle $\theta(X) \subseteq X$ by applying rules (a,b,c,d) with decreasing priority: (1°) if $width(X) < height(X)$, $\theta(X)$ has (a) greatest height, then (b) greatest width, and is located (c) topmost, then (d) leftmost; (2°)

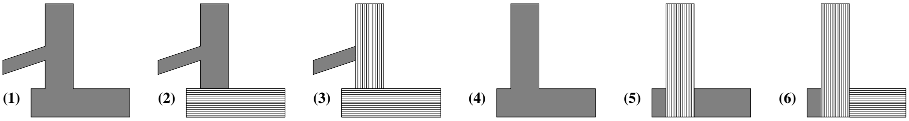


Fig. 3. $E = \mathbb{Z}^2$. (1) The set X, with $width(X) > height(X)$. (2) The rectangle $A = \theta(X)$ (shown with horizontal hatching) is selected with priority to width. (3) Now $width(X \setminus A) < height(X \setminus A) < \infty$, so the rectangle $B = \theta(X \setminus A)$ (shown with vertical hatching) is selected with priority to height; then $(\mathbf{1}_\theta \ltimes \mathbf{1}_\theta)(X) = \{A, B\}$ is invariant under further application of $\beta(\mathbf{1}_\theta \ltimes \mathbf{1}_\theta)$. (4) $Y = A \cup B$, with $width(Y) < height(Y) < \infty$. (5) The rectangle $C = \theta(Y)$ (shown with vertical hatching) is selected with priority to height. (6) Now $width(Y \setminus C) > height(Y \setminus C)$, so the rectangle $D = \theta(Y \setminus C)$ (shown with horizontal hatching) is selected with priority to width; then $(\mathbf{1}_\theta \ltimes \mathbf{1}_\theta)(Y) = \{C, D\}$, but $C \cup D \subset Y = A \cup B$.

if $width(X) \geq height(X)$, $\theta(X)$ has (a) greatest width, then (b) greatest height, and is located (c) leftmost, then (d) topmost. Consider now the *set shrinking* operator [7] $\mathbf{1}_\theta : X \mapsto \mathbf{1}_{\theta(X)}$. Then $\mathbf{1}_\theta$ is is $\mathcal{H}$-maximal for the set $\mathcal{H}$ of all rectangles; however the operator on $\psi : \mathcal{P}(E) \to \mathcal{P}(E) : X \mapsto \theta(X) \cup \theta\big(X \setminus \theta(X)\big)$ is not idempotent, see Figure 3.

In [8] we showed also that given a C-thinning κ and an adjunction (ε, δ), $\delta\kappa\varepsilon$ will be a C-thinning. This does not hold for max-thinnings. Indeed, the map $\mathsf{supp} : \Pi^*(E) \to \mathcal{P}(E) : \pi \mapsto \mathsf{supp}(\pi)$ is the lower adjoint of the map $\mathbf{1}_\bullet : \mathcal{P}(E) \to \Pi^*(E) : A \mapsto \mathbf{1}_A$ [5], and $\beta(\mathbf{1}_\theta \ltimes \mathbf{1}_\theta)$ is a max-thinning, but $\psi = \mathsf{supp} \cdot \beta(\mathbf{1}_\theta \ltimes \mathbf{1}_\theta) \cdot \mathbf{1}_\bullet$ is not idempotent.

5 Valuations on Partial Partitions

Let the space E be finite, so that $\Pi^*(E)$ is finite, and let $\underline{\blacktriangleleft}$ be a partial order relation on $\Pi^*(E)$ having $\boldsymbol{\emptyset}$ as least element. For $\pi_1, \pi_2 \in \Pi^*(E)$, we say that π_2 *covers* π_1 if $\pi_1 \blacktriangleleft \pi_2$ but there is no $\pi \in \Pi^*(E)$ with $\pi_1 \blacktriangleleft \pi \blacktriangleleft \pi_2$. The covering relation is written $\prec$, with possible variants like $\overset{x}{\prec}$. We call a *valuation* on $(\Pi^*(E), \underline{\blacktriangleleft})$ a map $f : \Pi^*(E) \to \mathbb{R}$ such that $f(\boldsymbol{\emptyset}) = 0$ and for all $\pi_1, \pi_2 \in \Pi^*(E)$, $\pi_1 \blacktriangleleft \pi_2 \;\Rightarrow\; f(\pi_1) < f(\pi_2)$; thus f is in fact $\Pi^*(E) \to \mathbb{R}^+$. Since $\Pi^*(E)$ is finite, the order $\underline{\blacktriangleleft}$ is determined by the covering relation $\prec$, and $f : \Pi^*(E) \to \mathbb{R}$ is a valuation iff $f(\boldsymbol{\emptyset}) = 0$ and for $\pi_1, \pi_2 \in \Pi^*(E)$, $\pi_1 \prec \pi_2 \;\Rightarrow\; f(\pi_1) < f(\pi_2)$. The valuation f is said to be *linear* if for any $\pi \in \Pi^*(E)$, $f(\pi) = \sum_{B \in \pi} f(\mathbf{1}_B)$, in other words if there is a map $g : \mathcal{P}(E) \setminus \{\emptyset\} \to \mathbb{R}$ such that for any $\pi \in \Pi^*(E)$, $f(\pi) = \sum_{B \in \pi} g(B)$. It is said to be *homogeneous* if it is invariant under any permutation of the points of E; equivalently, for any $\pi \in \Pi^*(E)$, $f(\pi)$ depends only on the histogram of sizes of all blocks of π, that is, the function $\mathbb{N} \to \mathbb{N} : n \mapsto \big|\{B \in \pi \mid |B| = n\}\big|$ (here $|X|$ denotes the size of a set X). It is both homogeneous and linear if there is a map $h : \mathbb{N} \setminus \{0\} \to \mathbb{R}$ such that for any $\pi \in \Pi^*(E)$, $f(\pi) = \sum_{B \in \pi} h(|B|)$.

Let us briefly describe the covering relation and linear valuations for the standard order $\leq$ on $\Pi^*(E)$. A similar analysis can easily be done for the well-composed order relations $\sqsubseteq$ and $\trianglelefteq$ (that admit also $\boldsymbol{\emptyset}$ as least element). For $\pi_1, \pi_2 \in \Pi^*(E)$, we say that π_2 *m-covers* (resp., *c-covers*) π_1 and write $\pi_1 \overset{m}{\prec} \pi_2$ (resp., $\pi_1 \overset{c}{\prec} \pi_2$) if π_2 is obtained from π_1 by merging two blocks (resp., by creating a new singleton block):

$$\pi_1 \overset{m}{\prec} \pi_2 \iff \exists\, B, C \in \pi_1,\ B \neq C,\ \pi_2 = \big(\pi_1 \setminus \{B, C\}\big) \cup \{B \cup C\} \ ,$$

$$\pi_1 \overset{c}{\prec} \pi_2 \iff \exists\, p \in E \setminus \mathsf{supp}(\pi_1),\ \pi_2 = \pi_1 \cup \big\{\{p\}\big\} \ .$$

We have $\pi_1 \prec \pi_2$ iff $\pi_1 \overset{m}{\prec} \pi_2$ or $\pi_1 \overset{c}{\prec} \pi_2$ [6]. Thus a linear valuation takes the form $f(\pi) = \sum_{B \in \pi} g(B)$, where $g : \mathcal{P}(E) \setminus \{\emptyset\} \to \mathbb{R}$ satisfies $g(\{p\}) > 0$ for all $p \in E$, and $g(B \cup C) > g(B) + g(C)$ for all disjoint $B, C \in \mathcal{P}(E) \setminus \{\emptyset\}$. A homogeneous linear valuation takes the form $f(\pi) = \sum_{B \in \pi} h(|B|)$, where $h : \mathbb{N} \setminus \{0\} \to \mathbb{R}$ satisfies $h(1) > 0$ and $h(m + n) > h(m) + h(n)$ for all $m, n > 0$. In a covering chain $\emptyset \prec \cdots \prec \pi$, we always have $|\mathsf{supp}(\pi)| - |\pi|$ m-coverings $\overset{m}{\prec}$ and $|\mathsf{supp}(\pi)|$ c-coverings $\overset{c}{\prec}$; assigning them respective costs $\mu, \gamma > 0$, we get the homogeneous linear valuation

$$f(\pi) = \mu\big(|\mathsf{supp}(\pi)| - |\pi|\big) + \gamma|\mathsf{supp}(\pi)| = (\mu + \gamma)|\mathsf{supp}(\pi)| - \mu|\pi| \ ; \qquad (2)$$

here $h(n) = (\mu + \gamma)n - \mu$ for all $n > 0$. For example, $\mu = \gamma = 1$ gives the *height* $2|\mathsf{supp}(\pi)| - |\pi|$ of π [6], with $h(n) = 2n - 1$.

Let us now apply valuations in image segmentation methodology. According to the maximality principle, $\sigma_{\mathsf{cr}}^F(A)$, the segmentation of $F : E \to T$ on $A \subseteq E$ following a criterion cr, is a maximal element of $\Pi^*(A, \mathcal{C}_{\mathsf{cr}}^F)$ for the chosen order $\underline{\blacktriangleleft}$ on $\Pi^*(E)$. We can modify that principle by requiring that $\sigma_{\mathsf{cr}}^F(A)$ is an element of $\Pi^*(A, \mathcal{C}_{\mathsf{cr}}^F)$ having maximum valuation; for the valuation f, we have:

$$\forall\, F \in T^E,\ \forall\, A \in \mathcal{P}(E), \quad f\big(\sigma_{\mathsf{cr}}^F(A)\big) = \max\big\{f(\pi) \mid \pi \in \Pi^*(A, \mathcal{C}_{\mathsf{cr}}^F)\big\}$$

This represents in fact a strengthening of the maximality principle; indeed, since $\pi_1 \blacktriangleleft \pi_2 \ \Rightarrow\ f(\pi_1) < f(\pi_2)$, an element of $\Pi^*(A, \mathcal{C}_{\mathsf{cr}}^F)$ with maximum valuation must necessarily be maximal.

Let now $\underline{\blacktriangleleft}$ be the standard order $\leq$. In connective segmentation, for each $F : E \to T$, $\mathcal{C}_{\mathsf{cr}}^F$ is a partial connection, that is, for every $A \in \mathcal{P}(E)$, $\Pi^*(A, \mathcal{C}_{\mathsf{cr}}^F)$ has a greatest element; then this greatest element has necessarily maximum valuation. Hence maximum valuation does not modify connective segmentation.

In Serra's method for "eliminating parasitic segmentation classes" [14,13], we start from an initial segmentation $\pi_0 \in \Pi^*(A, \mathcal{C}_{\mathsf{cr}}^F)$, eliminate "small parasitic" blocks, getting $\pi_1 \in \Pi^*(A, \mathcal{C}_{\mathsf{cr}}^F)$, then inflate the blocks of π_1, getting $\pi_2 \in \Pi^*(A, \mathcal{C}_{\mathsf{cr}}^F)$; ideally $\mathsf{supp}(\pi_0) \subseteq \mathsf{supp}(\pi_2) \subseteq A$. Clearly $\pi_0 \Subset \pi_1 \Subset \pi_2$, so from Serra's point of view, the modifications $\pi_0 \mapsto \pi_1 \mapsto \pi_2$ are two successive improvements. In the light of a valuation f, since $\pi_0 > \pi_1 < \pi_2$, we have $f(\pi_0) > f(\pi_1) < f(\pi_2)$, which is consistent with our view that the intermediate result π_1 is not satisfactory; now taking f as in (2), since π_2 has fewer blocks than π_0, and a larger or equal support, we get $f(\pi_2) > f(\pi_0)$, confirming that π_2 improves on π_1.

6 Conclusion and Perspectives

We have analysed image segmentation as the construction of an "optimal" partial partition with "homogeneous" blocks, where the optimality can mean either maximality for an order, or maximality of a valuation; the latter is generally a stronger requirement, except for connective segmentation, where the two are equivalent. Maximality for a well-composed order is preserved in compound segmentation, leading to max-thinnings on partial partitions. It seems that maximal valuation can lead to a segmentation eliminating "small parasitic" classes.

We have analysed several orders included in the standard order. There are other orders included in the building order, for example: $\pi_1 \underline{\blacktriangleleft} \pi_2$ iff each block of π_2 is a union of blocks of π_1, i.e., $\exists\, \pi,\ \pi_1 \supseteq \pi \sqsubseteq \pi_2$. Serra's work [13,14] suggests a new order relation on $\mathit{\Pi}^*(E)$, the *apportioning* order $\overline{\sqsubseteq}$, where $\pi_1 \mathrel{\overline{\sqsubseteq}} \pi_2$ iff $\pi_1 \Subset \pi_2$ and $\mathsf{supp}(\pi_1) = \mathsf{supp}(\pi_2)$. Consider further the *apportioning-inflating* order $\overline{\trianglelefteq}$, with $\pi_1 \mathrel{\overline{\trianglelefteq}} \pi_2$ iff $\pi_1 \Subset \pi_2$ and $\mathsf{supp}(\pi_1) \subseteq \mathsf{supp}(\pi_2)$ (briefly considered and written $\sqsubseteq$ in [13]), and the *extended* order $\overline{\leq}$, with $\pi_1 \mathrel{\overline{\leq}} \pi_2$ iff $\pi_1 \Subset \pi_2 \cap \mathcal{P}(\mathsf{supp}(\pi_1))$ and $\mathsf{supp}(\pi_1) \subseteq \mathsf{supp}(\pi_2)$. We then get the analogue of items 1 and 3 of Proposition 1 with $\overline{\sqsubseteq}, \overline{\trianglelefteq}, \overline{\leq}$ instead of $\sqsubseteq, \trianglelefteq, \leq$.

References

1. Meyer, F., Najman, L.: Segmentation, minimum spanning tree and hierarchies. In: Najman, L., Talbot, H. (eds.) Mathematical Morphology: From Theory to Applications, ch. 9, pp. 229–261. ISTE/J. Wiley & Sons (2010)
2. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(12), 1163–1173 (1996)
3. Ronse, C.: Toggles of openings, and a new family of idempotent operators on partially ordered sets. Applicable Algebra in Engineering, Communication and Computing 3, 99–128 (1992)
4. Ronse, C.: Set-theoretical algebraic approaches to connectivity in continuous or digital spaces. Journal of Mathematical Imaging and Vision 8(1), 41–58 (1998)
5. Ronse, C.: Partial partitions, partial connections and connective segmentation. Journal of Mathematical Imaging and Vision 32(2), 97–125 (2008), doi:10.1007/s10851-008-0090-5
6. Ronse, C.: Adjunctions on the lattices of partitions and of partial partitions. Applicable Algebra in Engineering, Communication and Computing 21(5), 343–396 (2010), doi:10.1007/s00200-010-0129-x
7. Ronse, C.: Idempotent block splitting on partial partitions, I: isotone operators. Order (to appear, 2011)
8. Ronse, C.: Idempotent block splitting on partial partitions, II: non-isotone operators. Order (to appear, 2011)
9. Ronse, C., Serra, J.: Algebraic foundations of morphology. In: Najman, L., Talbot, H. (eds.) Mathematical Morphology: From Theory to Applications, ch. 2, pp. 35–80. ISTE/J. Wiley & Sons (2010)
10. Serra, J.: Mathematical morphology for Boolean lattices. In: Serra, J. (ed.) Image Analysis and Mathematical Morphology. Theoretical Advances, vol. II, ch. 2, pp. 37–58. Academic Press, London (1988)
11. Serra, J.: Morphological segmentations of colour images. In: Ronse, C., Najman, L., Decencière, E. (eds.) Mathematical Morphology: 40 Years on, Computational Imaging and Vision, vol. 30, pp. 151–176. Springer, Dordrecht (2005)
12. Serra, J.: A lattice approach to image segmentation. Journal of Mathematical Imaging and Vision 24(1), 83–130 (2006)
13. Serra, J.: Ordre de la construction et segmentations hiérarchiques. Tech. rep., ESIEE/A2SI/IGM (2010)
14. Serra, J.: Grain buiding ordering. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 37–48. Springer, Heidelberg (2011)
15. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(7), 1132–1145 (2008)

Connective Segmentation Generalized to Arbitrary Complete Lattices

Seidon Alsaody[1,2] and Jean Serra[1]

[1] Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Equipe AS3I, ESIEE Paris (2, boulevard Blaise Pascal, Cité Descartes, BP 99, 93162 Noisy le Grand Cedex, France)

[2] Uppsala University, Department of Mathematics (P.O. Box 480, SE-751 06, Uppsala, Sweden)

Abstract. We begin by defining the setup and the framework of connective segmentation. Then we start from a theorem based on connective criteria, established for the power set of an arbitrary set. As the power set is an example of a complete lattice, we formulate and prove an analogue of the theorem for general complete lattices.

Secondly, we consider partial partitions and partial connections. We recall the definitions, and quote a result that gives a characterization of (partial) connections. As a continuation of the work in the first part, we generalize this characterization to complete lattices as well.

Finally we link these two approaches by means of a commutative diagram, in two manners.

Keywords: Connective segmentation, complete lattice, partial partition, block-splitting opening, commutative diagram.

1 Introduction

The theory of connective segmentation on sets, developed by, among others, Serra (see e.g. [8]) and Ronse (see e.g. [3]), has proved fruitful in image segmentation. In this article, we generalize this theory and concider connective segmentation on arbitrary complete lattices, rather than on the power set lattice $\mathcal{P}(E)$ of subsets of a set E. Apart from the theoretical value of such a generalization, it is also relevant in practice, as a number of important lattices are not of type $\mathcal{P}(E)$. Two major examples are

- viscous lattices, as described in [7]. The elements of a viscous lattice are the images of the subsets of a set E under a given dilation, ordered by inclusion. The smallest and greatest elements are $\emptyset \subseteq E$ and E, respectively, and for a non-empty family of subsets, the supremum is the union, whereas the infimum is the opening (corresponding to the dilation) of the intersection. Viscous lattices are atomistic, the atoms being not the points of E, as is true for $\mathcal{P}(E)$, but rather the images of such points under the dilation. For example, if $E = \mathbb{R}^n$, the structure element of the dilation may be a ball of fixed radius r. This gives a model of the physical world in the sense that atoms indeed are not singular points.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 61–72, 2011.

- the lattice of functions defined on an arbitrary set, and taking values on the extended real line $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$. While this lattice is indeed important, it is not even atomistic, and thus there is a motivation to extend the theory beyond atomisticity as well.

Section 1.2 gives the definitions and one of the main results to be generalized.

1.1 Notation

Throughout this paper, all sets considered that are not obtained from other sets as subsets (and that are usually denoted by E) are assumed to be non-empty, unless otherwise stated. Analogously, in all complete lattices that do not arise from a construction on other complete lattices (and that are usually denoted by $\mathcal{L}$) we assume that the smallest element, usually denoted by 0, and the greatest element, usually denoted by 1, do not coincide. Finally, the symbols $<$ and $\subset$ always denote strict inequality and strict inclusion, respectively.

1.2 Fundamentals of Connective Segmentation

First, we recall the definition of connections and connective criteria.

Definition 1. *[8] Let E and T be sets, and $\mathcal{F}$ a family of functions $f : E \to T$.*

1. *A* criterion *on $\mathcal{F}$ is a map $\sigma : \mathcal{F} \times \mathcal{P}(E) \to \{0, 1\}$ satisfying $\sigma(f, \emptyset) = 1$. The criterion σ is said to be* validated *on $(f, A) \in \mathcal{F} \times \mathcal{P}(E)$ whenever $\sigma(f, A) = 1$; otherwise it is* refuted *on (f, A).*
2. *A subset $\mathcal{C}$ of $\mathcal{P}(E)$ is a* connection *if*
   (a) $\emptyset \in \mathcal{C}$,
   (b) $\forall x \in E; \{x\} \in \mathcal{C}$, and
   (c) if $\{C_i\}_{i\in I} \subseteq \mathcal{C}$ for some index set I, and $\bigcap_{i\in I} C_i \neq \emptyset$, then $\bigcup_{i\in I} C_i \in \mathcal{C}$.
3. *A criterion σ on $\mathcal{F}$ is* connective *if for each $f \in \mathcal{F}$, the set $\sigma_f = \{A \in \mathcal{P}(E); \sigma(f, A) = 1\}$ is a connection.*

The elements of $\mathcal{C}$ are said to be *$\mathcal{C}$-connected*, or *connected* if the connection is clear from the context. A *connected component* of a set $A \subseteq E$ is an element $C \in \mathcal{C} \cap \mathcal{P}(A)$, $C \neq \emptyset$, such that there exists no $D \in \mathcal{C}$ that satisfies $C \subset D \subseteq A$; if $A = E$, such sets will simply be called connected components.

Next we define partitions and segmentation.

Definition 2. *A* partition *of E is a map $\pi : E \to \mathcal{P}(E)$ that satisfies*

1. *$\forall x \in E; x \in \pi(x)$, and*
2. *$\forall x, y \in E$ such that $\pi(x) \cap \pi(y) \neq \emptyset$ it holds that $\pi(x) = \pi(y)$.*

When confusion is improbable, we will denote the image of a partition π, i.e the collection of all its classes, by π as well. Thus the notation $C \in \pi$ means that C is a class of π.

The collection of all partitions of a set E, denoted $\Pi(E)$, is known to be a complete lattice under the *refinement order*, defined as follows. For $\pi, \pi' \in \Pi(E)$, $\pi \leq \pi'$ if and only if $\pi(x) \subseteq \pi'(x)$ for each $x \in E$, i.e. if and only if each class of π is contained in a class of π'. The smallest element (or empty supremum) in this lattice is the partition $I; I(x) = \{x\}$ for all $x \in E$, while the greatest partition (or empty infimum) is $1; 1(x) = E$ for all $x \in E$. The infimum π of a non-empty family $\{\pi_i\}_{i \in I}$ of partitions, where I is an index set, is the partition whose classes are the intersections

$$\pi(x) = \bigcap_{i \in I} \pi_i(x) \tag{1}$$

while the supremum π' of $\{\pi_i\}_{i \in I}$ is the smallest partition such that for each $x \in E$, $\pi_i(x) \subseteq \pi'(x)$ for all $i \in I$.[1] Partitions are the corner stone of segmentation.

Definition 3. *[8] Given $f \in \mathcal{F}$ where $\mathcal{F}$ is a family of functions from a set E to a set T, and given $A \in \mathcal{P}(E)$ and a criterion σ on $\mathcal{F}$, let $\Pi(A, \sigma_f)$ be the family of all partitions π on A such that σ is validated on each class of π. The criterion σ is said to* segment *f (over A) if*

1. *$\forall x \in E, \sigma(f, \{x\}) = 1$, and*
2. *the family $\Pi(A, \sigma_f)$ is closed under the supremum of partitions.*

In that case, the supremum of $\Pi(A, \sigma_f)$ is the segmentation of f on A by σ.

Remark 1. Closure under the supremum in item 2 is equivalently expressed by saying that $\Pi(A, \sigma_f)$ is a dual Moore family in $\Pi(A)$. When the set A is not specified, the segmentation is understood to be over E (as is the case in the theorem below). Note that item 1 is the special case of item 2 when taking the empty supremum of partitions, thus making item 1 redundant. Nevertheless, we use this formulation to prepare for the generalized version of the definition.

We aim at generalizing the following result by Serra.

Theorem 1. *[8] Given two sets E and T, let $\mathcal{F}$ be a family of functions $f : E \to T$, and σ a criterion on $\mathcal{F}$. Then σ is connective if and only if σ segments all $f \in \mathcal{F}$.*

From [8] we know that in this case, for each $f \in \mathcal{F}$ the segmentation obtained is the partition of E into its σ_f-connected components.[2]

2 Generalization to Arbitrary Complete Lattices

In this section, we will consider arbitrary complete lattices $\mathcal{L} = (\mathcal{L}, \leq)$ sup-generated by a set $S \subseteq \mathcal{L}$. (This indeed holds for all complete lattices $\mathcal{L}$ by

[1] This is a formal definition. A way to construct the supremum is by *chaining*; see [2].

[2] This will indeed be the case in every generalization of the theorem, as the reader will see in the coming sections.

choosing $S = \mathcal{L} \setminus \{0\}$, but often more interesting choices can be made. If, for example, $\mathcal{L}$ is atomistic, then S can be chosen to be the set of all atoms.) We always assume $0 \notin S$. An important property of a sup-generating set is that for any $x, y \in \mathcal{L}$, it holds that[3]

$$(\forall s \in S; s \leq x \Rightarrow s \leq y,) \Longleftrightarrow x \leq y \ . \tag{2}$$

2.1 Adapting the Setting

We begin by generalizing the definitions.

Definition 4. *Let $\mathcal{L}$ be a complete lattice sup-generated by a subset $S \subseteq \mathcal{L}$, T an arbitrary set, and $\mathcal{F}$ a family of functions $f : S \to T$.*

1. *A* criterion *on $\mathcal{F}$ is a map $\sigma : \mathcal{F} \times \mathcal{L} \to \{0, 1\}$ satisfying $\sigma(f, 0) = 1$. The criterion σ is said to be* validated *on $(f, l) \in \mathcal{F} \times \mathcal{L}$ whenever $\sigma(f, l) = 1$; otherwise it is* refuted *on (f, l).*
2. *[5] A subset $\mathcal{C}$ of $\mathcal{L}$ is an S-*connection *if it satisfies*
   (a) $0 \in \mathcal{C}$,
   (b) $S \subseteq \mathcal{C}$, and
   (c) if $\{c_i\}_{i \in I} \subseteq \mathcal{C}$ for some index set I, and $\bigwedge_{i \in I} c_i \neq 0$, then $\bigvee_{i \in I} c_i \in \mathcal{C}$.
3. *A criterion σ on $\mathcal{F}$ is S-*connective *if for each $f \in \mathcal{F}$, the set $\sigma_f = \{l \in \mathcal{L}; \sigma(f, l) = 1\}$ is an S-connection.*

Note that the domain of the functions is S and not $\mathcal{L}$; indeed S generalizes the set E of Definition 1, and $\mathcal{L}$ generalizes $\mathcal{P}(E)$. (In [5], an S-connection is just called a *connection*.)

For simplicity, when the function f is clear from context, we will say that σ is validated on $l \in \mathcal{L}$, meaning that $\sigma(f, l) = 1$. An S-connected component of an element $l \in \mathcal{L}$ greater than or equal to a given $s \in S$ is an element $c \in \mathcal{C}$ such that $s \leq c \leq l$ and there exists no $d \in \mathcal{C}$ such that $c < d \leq l$; due to Definition 4.2.(c), this is precisely $\bigvee\{c \in \mathcal{C}; s \leq c \leq l\}$.

Next is the definition of partitions on complete lattices.

Definition 5. *[1] Let $\mathcal{L}$ be a complete lattice sup-generated by a subset $S \subseteq \mathcal{L}$. An S-*partition *on $\mathcal{L}$ is a map $\pi : S \to \mathcal{L}$ that satisfies*

1. *$\forall s \in S; s \leq \pi(s)$, and*
2. *$\forall s, t \in S$ such that $\pi(s) \wedge \pi(t) \neq 0$ it holds that $\pi(s) = \pi(t)$.*

Ordered by refinement, in analogy with the case of $\Pi(E)$, the set $\Pi_S(\mathcal{L})$ of S-partitions on $\mathcal{L}$ is a complete lattice. As before, we will use the same notation for a partition and its image (which we continue to call the set of its classes, even though the term no longer bears its original meaning), and the prefix S will sometimes be dropped when the set S is clear from context. Segmentation is defined as follows.

[3] Thus this holds for $S = \mathcal{L} \setminus \{0\}$ as a special case; indeed, the strength of the notion of a sup-generating set is that any such set S satisfies this property.

Definition 6. *Let $\mathcal{L}$ be a complete lattice sup-generated by a subset $S \subseteq \mathcal{L}$. Given $f \in \mathcal{F}$ where $\mathcal{F}$ is a family of functions from S to a set T, and a criterion σ on $\mathcal{F}$, let $\mathit{\Pi}_S(\mathcal{L}, \sigma_f)$ be the family of all S-partitions π of $\mathcal{L}$ such that σ is validated on each class of π. The criterion σ is said to S-*segment *f if*

1. *$\forall s \in S, \sigma(f, s) = 1$, and*
2. *the family $\mathit{\Pi}_S(\mathcal{L}, \sigma_f)$ is closed under the supremum of S-partitions.*

*In that case the supremum of $\mathit{\Pi}_S(\mathcal{L}, \sigma_f)$ is the S-*segmentation of *f* by *σ.*

Remark 2. For a set E, the lattice $\mathcal{P}(E)$ is sup-generated by the set of singleton sets, i.e. the points of E. Setting $\mathcal{L} = \mathcal{P}(E)$ and $S = \{\{x\}; x \in E\}$ we retrieve the original definitions from the generalized ones upon identifying $E = \{\{x\}; x \in E\}$.

Remark 3. In general lattices, the empty supremum of partitions is not the map $\pi : S \to \mathcal{L}, \pi(s) = s$, as this is not necessarily a partition (see, for instance, the lattice in Example 1). Thus item 2 does not imply item 1. Lemma 1 below will describe the empty supremum from a connectivity point of view.

2.2 Main Result

We now aim to generalize Theorem 1, for which we need the following

Lemma 1. *Let $\mathcal{L}$ be a complete lattice sup-generated by a subset $S \subseteq \mathcal{L}$. Let $\mathit{\Pi}_S(\mathcal{L})$ be the complete lattice of all S-partitions on $\mathcal{L}$, and denote by π its smallest element. Then for any S-connection $\mathcal{C}$ on $\mathcal{L}$, $\pi(s) \in \mathcal{C}$ for all $s \in S$.*

Proof. Given an S-connection $\mathcal{C}$, consider the map $\pi_C : S \to \mathcal{L}$ defined by[4]

$$\pi_C(s) = \bigvee \{c \in \mathcal{C}; s \leq c \leq \pi(s)\} \tag{3}$$

for all $s \in S$. Then as π is an S-partition, it follows that π_C is an S-partition. Moreover, $\pi_C(s) \in \mathcal{C}$ for all $s \in S$ by Definition 4.2.(c), and in the lattice $\mathit{\Pi}_S(\mathcal{L})$ we have $\pi_C \leq \pi$, thus $\pi_C = \pi$ since π is the smallest S-partition on $\mathcal{L}$. □

Theorem 2. *Let $\mathcal{L}$ be a complete lattice sup-generated by a subset $S \subseteq \mathcal{L}$. Let $\mathcal{F}$ be a family of functions $f : S \to T$ where T is an arbitrary set, and let σ be a criterion on $\mathcal{F}$.*

1. *If σ is S-connective, then σ S-segments all $f \in \mathcal{F}$.*
2. *If $\mathcal{L}$ is atomistic, and S is the set of all atoms of $\mathcal{L}$, it conversely holds that if σ S-segments all $f \in \mathcal{F}$, then σ is S-connective.*

Proof. Throughout the proof, let f be any element of $\mathcal{F}$. Let $\sigma_f = \{l \in \mathcal{L}; \sigma(f, l) = 1\}$, and let $\mathit{\Pi}_S(\mathcal{L}, \sigma_f)$ be the collection of all S-partitions all of whose classes belong to σ_f. We start by proving item 1.

[4] This definition is equivalent to saying that π_C assigns to each $s \in S$ the connected component of $\pi(s)$ greater than or equal to it.

Assume that σ is S-connective. By Definition 4.2.(b), the first item of Definition 6 is satisfied. Definition 6.2. is satisfied for the empty supremum (i.e. smallest S-partition) of $\Pi_S(\mathcal{L},\sigma_f)$ since by Lemma 1 the empty supremum of $\Pi_S(\mathcal{L})$ is in $\Pi_S(\mathcal{L},\sigma_f)$. Next, let $\{\pi_i\}_{i\in I}$, for some index set I, be a non-empty family in $\Pi_S(\mathcal{L},\sigma_f)$ with $\pi = \bigvee_{i\in I}\pi_i$. Let $\pi_C : S \to \sigma_f$ be defined by

$$\pi_C(s) = \bigvee\{c \in \sigma_f; s \le c \le \pi(s)\} \tag{4}$$

for all $s \in S$. It is readily checked that π_C is a partition. Now, $\pi_C \le \pi$, and for all $i \in I$ and $s \in S$, $\pi_i(s) \le \pi_C(S)$ since $\pi_i(s) \in \sigma_f$ and $s \le \pi_i(s) \le \pi(s)$. Hence π_C is an upper bound of $\{\pi_i\}_{i\in I}$, and π being the least such, we have $\pi \le \pi_C$ and altogether $\pi = \pi_C$. Thus $\Pi_S(\mathcal{L},\sigma_f)$ is also closed under non-empty suprema. This proves item 1.

As for the converse, by Definitions 4.1 and 6.1, $\sigma(f,l) = 1$ for all $l \in \{0\}\cup S$. Take any $\{c_i\}_{i\in I} \subseteq \sigma_f$ satisfying $\bigwedge_{i\in I} c_i \ne 0$. Since S is sup-generating, this implies that $\exists s_0 \in S; s_0 \le \bigwedge_{i\in I} c_i$. Each c_i induces a map $\pi_i : S \to \mathcal{L}$ defined by

$$\begin{cases} \pi_i(s) = c_i & \text{if } s \le c_i \text{ and} \\ \pi_i(s) = s & \text{otherwise .} \end{cases} \tag{5}$$

If each s is an atom, then indeed each π_i is a partition, and clearly we have $\pi_i \in \Pi_S(\mathcal{L},\sigma_f)$ for all $i \in I$. Since σ S-segments all $f \in \mathcal{F}$, it follows that $\pi = \bigvee_{i\in I}\pi_i \in \Pi_S(\mathcal{L},\sigma_f)$. Define $x = \bigvee_{i\in I} c_i$. We need show that $x \in \sigma_f$. For any $i \in I$, the definition of π_i implies that $\pi_i(s_0) = c_i$ and hence $\pi_i(s_0) \le x$ for all $i \in I$. Now define the partition π' by

$$\begin{cases} \pi'(s) = x & \text{if } s \le x \text{ and} \\ \pi'(s) = s & \text{otherwise} \end{cases} \tag{6}$$

for all $s \in S$. For all $i \in I$ it holds that $\pi_i \le \pi'$, and as $\pi = \bigvee_{i\in I}\pi_i$, we have $\pi \le \pi'$, and thus $\pi(s_0) \le \pi'(s_0) = x$.

Since $\pi_i \le \pi$, we have $c_i = \pi_i(s_0) \le \pi(s_0)$ for all $i \in I$. By the definition of the supremum this implies that $x = \bigvee_{i\in I} c_i \le \pi(s_0)$. Altogether $x = \pi(s_0)$, whereby $x \in \sigma_f$ since $\Pi_S(\mathcal{L},\sigma_f)$ is closed under suprema. Thus σ_f is an S-connection and the proof is complete. □

The following example demonstrates that item 2 of the theorem is in general false when atomisticity is not required.

Example 1. Let $\mathcal{L}$ be the set $\{0, 1, s_0, s_1, s_2, s_3, t\}$ and define the partial order $\le$ on $\mathcal{L}$ by

$$\begin{tikzcd} 0 \arrow[r] & s_0 \arrow[r] \arrow[dr] & s_1 \arrow[r] \arrow[dr] & s_3 \arrow[r] & 1 \\ & & s_2 \arrow[r] & t \arrow[ur] & \end{tikzcd} \tag{7}$$

where for $l \in \mathcal{L}$ and $m \in \mathcal{L}$, $l \le m$ if and only if there exists a directed path from l to m. It is easily checked that $\mathcal{L}$ is a lattice, hence a complete lattice by finiteness, and that it is sup-generated by the set $S = \{s_i, i = 0, 1, 2, 3\}$. It is however not atomistic as e.g. s_0 and s_1 necessarily belong to each sup-generating set. Let T be any set, and $\mathcal{F}$ any family of functions from S to T. Define the criterion σ on $\mathcal{F}$ by

$$\begin{cases} \sigma(f, l) = 0 & \text{if } l = t \text{ and} \\ \sigma(f, l) = 1 & \text{otherwise} \end{cases} \tag{8}$$

for all $f \in \mathcal{F}$. Since $t = s_1 \vee s_2$, and $s_1 \wedge s_2 \neq 0$, this implies that σ is not connective. However, by the structure of $\mathcal{L}$ the only possible S-partition on $\mathcal{L}$ is the one whose unique class is 1, and as $\sigma(f, 1) = 1$, we get that σ S-segments f.

Thus we conclude that Theorem 1 generalizes completely to the atomistic setting, but in general not farther. The approach of Section 3 will, as it turns out, in fact mend this issue.

3 Partial Connections and Partial Partitions

We now extend this study in another direction, by looking at operators that act on partitions. More specifically, we study a certain type of openings on partial partitions, and their relation to partial connections (concepts to be defined). This follows the work of Ronse, who has established this theory for the case of the power set lattice in [4]. We begin by the definitions for that case.

Definition 7. *[3] Let E be a set.*

1. *A subset $\mathcal{C}$ of $\mathcal{P}(E)$ is a* partial connection *on E if it satisfies*
   (a) $\emptyset \in \mathcal{C}$, and
   (b) if $\{C_i\}_{i \in I} \subseteq \mathcal{C}$ for some index set I, and $\bigcap_{i \in I} C_i \neq \emptyset$, then $\bigcup_{i \in I} C_i \in \mathcal{C}$.
2. *A* partial partition *of E is a map $\pi : E \to \mathcal{P}(E)$ that satisfies*
   (a) $\forall x \in E; \pi(x) = \emptyset$ or $x \in \pi(x)$, and
   (b) $\forall x, y \in E; x \in \pi(y) \Rightarrow \pi(x) = \pi(y)$.

It is seen that every connection is a partial connection, and by [3] every partition is a partial partition. As in the case of partitions, the notation $C \in \pi$ means that C is a class of π. Definition 7 implies that singleton sets are not generally assumed to be connected, and that the connected components of a partial connection and the classes of a partial partition do not necessarily cover E. In [4], one motivation of using partiality is that in computations, considering singleton sets to be a priori connected becomes impractical. In general, this offers a great deal of flexibility to the theory, as we will see.

The collection of partial connections on E, denoted by $\Gamma^*(E)$, is a complete lattice under the inclusion order: for $\mathcal{C}, \mathcal{D} \in \Gamma^*(E)$, $\mathcal{C} \leq \mathcal{D} \iff \forall C \in \mathcal{C}, C \in \mathcal{D}$; the set of partial partitions on E is denoted $\Pi^*(E)$; it is a complete lattice under the refinement order previously defined for partitions. [3] The greatest element in this lattice is the partition $1; 1(x) = E$ for all $x \in E$, while the smallest partial partition is $0; 0(x) = \emptyset$ for all $x \in E$.

The notions of *partially connective criteria* and *partial segmentation*, in which a criterion *segments* a function *partially*, are obtained by replacing connections and partitions in Definitions 1.3 and 3.2 by partial connections and partial partitions, and disregarding Definition 3.1. (about singleton sets). A detailed study of partial connections and partial partitions is conducted in [3]. We move on to a class of operators on $\Pi^*(E)$, defined in two steps.

Definition 8. *[4] Let E be a set.*

1. *A map $\psi : \mathcal{P}(E) \to \Pi^*(E)$ is* set splitting *if $\psi(A) \in \Pi^*(A)$, i.e. if $\psi(A)(x) \subseteq A$ for all $x \in E$.*
2. *Given a set splitting map ψ, the* class splitting operator induced by ψ *is the operator $\psi^* : \Pi^*(E) \to \Pi^*(E)$ such that for each partial partition $\pi \in \Pi^*(E)$, $\psi^*(\pi) = \bigvee_{C \in \pi} \psi(C)$.*

Note that the supremum in item 2 of the definition is in fact the partial partition whose classes are the images under ψ of the classes of π.

From [4] one learns that ψ is set splitting if and only if for all $A \subseteq E$ and $x \in E \setminus A$, $\psi(A)(x) = \emptyset$. In [4] class splitting operators are called *block splitting.*

3.1 Openings on Partial Partitions

An opening (say, ω) on a complete lattice is an anti-extensive ($\omega(x) \leq x$), idempotent and order-preserving operator on it. The set of openings on a complete lattice $\mathcal{L}$ is a subset of the complete lattice of operators on $\mathcal{L}$. It exhibits additional structure by the following well-known fact.

Lemma 2. *[6] Let $\mathcal{L}$ be a complete lattice, and $\mathcal{L}^{\mathcal{L}}$ the complete lattice of operators on $\mathcal{L}$. The subset of $\mathcal{L}^{\mathcal{L}}$ consisting of all openings on $\mathcal{L}$ is a dual Moore family[5] in $\mathcal{L}^{\mathcal{L}}$. It is thus a complete lattice under the order induced by the order on $\mathcal{L}^{\mathcal{L}}$.*

In this lattice, the supremum resp. the infimum of a non-empty family P of openings is the supremum of P in $\mathcal{L}^{\mathcal{L}}$ resp. the greatest opening smaller than or equal to the infimum of P in $\mathcal{L}^{\mathcal{L}}$; the smallest element in the lattice of openings is the map $l \mapsto 0$ for all $l \in \mathcal{L}$, and the greatest element is the identity map $l \mapsto l$ for all $l \in \mathcal{L}$.

There is indeed a greatest opening smaller than or equal to a given element in $\mathcal{L}^{\mathcal{L}}$, by definition of a dual Moore family; it is for the same reason that the supremum in $\mathcal{L}^{\mathcal{L}}$ of a family of openings is an opening.

The following result by Ronse is central in the characterization of partial connections in terms of class splitting openings.

Proposition 1. *[4] The set $\Omega(E)$ of class splitting openings on $\Pi^*(E)$ for a set E is a complete sublattice of the lattice of openings on $\Pi^*(E)$. It is isomorphic to the lattice $\Gamma^*(E)$ of partial connections. The isomorphism is given by the map*

$$\lambda : \Gamma^*(E) \to \Omega(E); \mathcal{C} \mapsto \mathcal{C}^*_\bullet \tag{9}$$

where for all $\pi \in \Pi^*(E)$, $\mathcal{C}^*_\bullet(\pi)(x) = \bigvee\{C \in \mathcal{C}; x \in C \subseteq \pi(x)\}$ for all $x \in E$.

[5] See Remark 1.

Put simply, λ maps each partial connection $\mathcal{C}$ to the (class splitting) opening that splits the classes of a partial partition into their $\mathcal{C}$-connected components.

3.2 Generalizing Ronse's Result

Following the lines of our treatment of partitions and connections on complete lattices, Definition 7 generalizes readily. Note that, in contrast to the generalization of the concept of a connection (Definition 4.2), the partial counterpart is independent of any sup-generating set. Indeed, the dependence of Definition 4.2 on a sup-generating set lies in its second item, which does not appear here, generalizing the aforementioned fact that in the power set lattice, singleton sets are now not a priori assumed connected.

Definition 9. *1. Let $\mathcal{L}$ be a complete lattice. A subset $\mathcal{C}$ of $\mathcal{L}$ is a* partial connection *if it satisfies*

(a) $0 \in \mathcal{C}$*, and*

(b) if $\{c_i\}_{i \in I} \subseteq C$ *for some index set* I*, and* $\bigwedge_{i \in I} c_i \neq 0$*, then* $\bigvee_{i \in I} c_i \in \mathcal{C}$*.*

2. Let $\mathcal{L}$ be a complete lattice sup-generated by a subset $S \subseteq \mathcal{L}$. A partial S-partition *on $\mathcal{L}$ is a map $\pi : S \to \mathcal{L}$ that satisfies*

(a) $\forall s \in S; \pi(s) = 0$ or $s \leq \pi(s)$*, and*

(b) $\forall s, t \in S$ *such that* $s \leq \pi(t)$ *it holds that* $\pi(s) = \pi(t)$*.*

The sets of partial connections and partial S-partitions on $\mathcal{L}$, henceforth denoted by $\Gamma^*(\mathcal{L})$ and $\Pi^*_S(\mathcal{L})$, respectively, are complete lattices by a straight-forward proof. *Partial S-segmentation*, where a criterion *S-segments* a function *partially*, and *partially connective criteria* are defined in an analogous manner to that described above Definition 8 for the power set lattice.

Example 2. If $\mathcal{L}$ is a complete lattice sup-generated by $S \subseteq \mathcal{L}$, then for each $m \in \mathcal{L}$, the function $1_m : S \to \mathcal{L}$ defined by $1_m(s) = m$ if $s \leq m$, and $1_m(s) = 0$ otherwise, is a partial S-partition with m as unique non-zero class.

The class splitting operators become as follows in the general setting.

Definition 10. *Let $\mathcal{L}$ be a complete lattice sup-generated by a subset $S \subseteq \mathcal{L}$.*

1. *A map $\psi : \mathcal{L} \to \Pi^*_S(\mathcal{L})$ is* element splitting *if $\psi(l) \leq 1_l$ for all $l \in \mathcal{L}$; equivalently, if $\forall l \in \mathcal{L}; \forall s \in S; \psi(l)(s) \leq l$.*
2. *Given an element splitting map ψ, the* class splitting operator induced by ψ *is the operator $\psi^* : \Pi^*_S(\mathcal{L}) \to \Pi^*_S(\mathcal{L})$ such that for each partial S-partition $\pi \in \Pi^*_S(\mathcal{L})$, $\psi^*(\pi) = \bigvee_{c \in \pi} \psi(c)$.*

Now we are ready to generalize Proposition 1.

Theorem 3. *Let $\mathcal{L}$ be a complete lattice sup-generated by a subset $S \subseteq \mathcal{L}$. Let $\Omega_S(\mathcal{L})$ be the set of all class splitting openings on $\Pi^*_S(\mathcal{L})$. Then $\Omega_S(\mathcal{L})$ is a complete sublattice of the lattice of openings on $\Pi^*_S(\mathcal{L})$. As a complete lattice, it is isomorphic to the lattice $\Gamma^*(\mathcal{L})$ of partial connections on $\mathcal{L}$, via the isomorphism*

$$\lambda : \Gamma^*(\mathcal{L}) \to \Omega_S(\mathcal{L}); \mathcal{C} \mapsto \mathcal{C}^*_\bullet \tag{10}$$

where for all $\pi \in \Pi^*_S(\mathcal{L})$, $\mathcal{C}^*_\bullet(\pi)(s) = \bigvee\{c \in \mathcal{C}; s \leq c \leq \pi(s)\}$ for all $s \in S$.

The long, technical proof leading to this result follows the lines of the proof of Proposition 1, given in [4], and we here give a brief sketch due to space limits.

Proof. The smallest opening on $\Pi^*_S(\mathcal{L})$ is $\mathbf{0} : \pi \mapsto 0$, which is class splitting. The greatest opening on $\Pi^*_S(\mathcal{L})$ is the identity map, which is class splitting as induced by the element splitting map $m \mapsto 1_m$. Consider next a non-empty family of class splitting openings $\{\psi^*_i\}_{i\in I}$ for an index set I. Each ψ^*_i is shown to be induced by a unique, order-preserving element splitting map ψ_i. By Lemma 2, the supremum $\bigvee_{i\in I}(\psi^*_i)$ is an opening, which is in fact class splitting. The infimum in the lattice of openings is described in Lemma 2. One deduces that it is class splitting from the ψ_i being order preserving. This proves the first part. Given $\mathcal{C} \in \Gamma^*(\mathcal{L})$, the operator $\mathcal{C}^*_\bullet : \Pi^*_S(\mathcal{L}) \to \Pi^*_S(\mathcal{L})$ is a class splitting opening, induced by the map that splits an element of $\mathcal{L}$ into its $\mathcal{C}$-connected components. Thus λ is well-defined, and an isomorphism by technical considerations. □

We moreover give the following result, which is the analogue of Theorem 2 in the partial setup. The special case for the power set lattice was proven in [3].

Lemma 3. *Let $\mathcal{L}$ be a complete lattice sup-generated by a subset $S \subseteq \mathcal{L}$, and let $\mathcal{C} \subseteq \mathcal{L}$ be a subset. Then $\mathcal{C}$ is a partial connection if and only if the set $\Pi^*_S(\mathcal{L},\mathcal{C})$ of partial S-partitions all whose classes are in $\mathcal{C}$ is closed under arbitrary suprema.*

Corollary 1. *Let $\mathcal{L}$ be a complete lattice sup-generated by a subset $S \subseteq \mathcal{L}$, $\mathcal{F}$ a family of functions $f : S \to T$ for a set T, and σ a criterion on $\mathcal{F}$. Then σ is partially connective if and only if σ S-segments all $f \in \mathcal{F}$ partially.*

Note how partiality circumvents the need of atomisticity in order for the result to be fully valid as no longer all $s \in S$ are required to be connected.

4 Relating Ronse's Approach to Serra's

In this section we will state and prove a relation between the characterization of (partially) segmenting criteria of Corollary 1 (Theorem 2), and that of class splitting openings of Theorem 3; this will be done by means of a commutative diagram. We give the link in both the partial and non-partial setup, and we state it directly for general lattices; of course, it then also holds for the special case of the power set lattice. To the authors' knowledge, such a link has not been presented before in any setting.

For connective segmentation in the non-partial sense, we have the following

Theorem 4. *Let $\mathcal{L}$ be a complete lattice, sup-generated by a subset $S \subseteq \mathcal{L}$, T a set, and $\mathcal{F}$ a family of functions $f : S \to T$. Define $\Sigma_S(\mathcal{F},\mathcal{L})$ to be the set of all S-connective criteria on $\mathcal{F}$, $\Pi_S(\mathcal{L})$ the complete lattice of S-partitions on $\mathcal{L}$, $\Pi^*_S(\mathcal{L})$ the complete lattice of partial S-partitions on $\mathcal{L}$, $\Gamma^*(\mathcal{L})$ the complete lattice of partial connections on $\mathcal{L}$, and $\Omega_S(\mathcal{L})$ the complete lattice of class splitting openings on $\Pi^*_S(\mathcal{L})$. Let moreover*

- *for each $f \in \mathcal{F}, \phi_f : \varSigma_S(\mathcal{F}, \mathcal{L}) \to \varPi_S(\mathcal{L})$ be the map that assigns to each S-connective criterion σ its S-segmentation of f, and*
- *$\lambda : \varGamma^*(\mathcal{L}) \to \varOmega_S(\mathcal{L})$ be the lattice isomorphism that assigns to each partial connection $\mathcal{C}$ the class splitting opening $\mathcal{C}^*_\bullet$.*

Then for each $f \in \mathcal{F}$ there exist maps μ_f and κ such that following diagram commutes

$$\begin{tikzcd}
\varSigma_S(\mathcal{F}, \mathcal{L}) \arrow[r, "\phi_f"] \arrow[d, "\mu_f"] & \varPi_S(\mathcal{L}) \\
\varGamma^*(\mathcal{L}) \arrow[r, "\lambda"'] & \varOmega_S(\mathcal{L}) \arrow[u, "\kappa"]
\end{tikzcd} \quad . \tag{11}$$

In other words we have $\phi_f(\sigma) = \kappa\lambda\mu_f(\sigma)$ for all $f \in \mathcal{F}$ and $\sigma \in \varSigma_S(\mathcal{F}, \mathcal{L})$. Specifically,

- *μ_f is the map that assigns to each S-connective criterion σ the S-connection $\{c \in \mathcal{L}; \sigma(f, c) = 1\}$, and*
- *κ is the map that assigns to each class splitting opening γ^* the partial S-partition $\gamma^*(1)$, where 1 is the greatest S-partition on $\mathcal{L}$.*

Proof. Given $f \in \mathcal{F}$, take any criterion $\sigma \in \varSigma_S(\mathcal{F}, \mathcal{L})$. Then $\mu_f(\sigma) = \{c \in \mathcal{L}; \sigma(f, c) = 1\}$, and composing μ_f with λ and κ gives $\kappa\lambda\mu_f(\sigma) = (\mu_f(\sigma))^*_\bullet(1)$, where for all $s \in S$,

$$(\mu_f(\sigma))^*_\bullet(1)(s) = \bigvee\{c \in \mu_f(\sigma); s \leq c \leq 1\} = \bigvee\{c \in \mu_f(\sigma); s \leq c\} \ , \tag{12}$$

i.e. $(\mu_f(\sigma))^*_\bullet(1)(s)$ is equal to the $\mu_f(\sigma)$-connected component greater than or equal to s, if it exists, and otherwise it is equal to 0. Since $\mu_f(\sigma)$ is a partial connection, $(\mu_f(\sigma))^*_\bullet(1)$ is a partial S-partition all of whose classes belong to $\mu_f(\sigma)$. It is moreover an S-partition, since $\mu_f(\sigma)$ is an S-connection. This is the largest S-partition all of whose classes belong to $\mu_f(\sigma)$. Hence it is the S-segmentation of f by σ, which completes the proof. □

Next, we replace connections and partitions by partial connections and partial partitions, respectively, to get the following

Theorem 5. *Let $\mathcal{L}$ be a complete lattice, sup-generated by a subset $S \subseteq \mathcal{L}$, T a set, and $\mathcal{F}$ a family of functions $f : S \to T$. Define $\varSigma^*(\mathcal{F}, \mathcal{L})$ to be the set of all partially connective criteria on $\mathcal{F}$, $\varPi^*_S(\mathcal{L})$ the complete lattice of partial S-partitions on $\mathcal{L}$, $\varGamma^*(\mathcal{L})$ the complete lattice of partial connections on $\mathcal{L}$, and $\varOmega_S(\mathcal{L})$ the complete lattice of class splitting openings on $\varPi^*_S(\mathcal{L})$. Let moreover*

- *for each $f \in \mathcal{F}, \phi'_f : \varSigma^*(\mathcal{F}, \mathcal{L}) \to \varPi^*_S(\mathcal{L})$ be the map that assigns to each partially connective criterion σ its partial S-segmentation of f, and*
- *$\lambda : \varGamma^*(\mathcal{L}) \to \varOmega_S(\mathcal{L})$ be the lattice isomorphism that assigns to each partial connection $\mathcal{C}$ the class splitting opening $\mathcal{C}^*_\bullet$.*

Then for each $f \in \mathcal{F}$ there exist maps μ'_f and κ such that following diagram commutes

$$\begin{tikzcd}
\Sigma^*(\mathcal{F},\mathcal{L}) \arrow[r, "\phi'_f"] \arrow[d, "\mu'_f"] & \Pi^*_S(\mathcal{L}) \\
\Gamma^*(\mathcal{L}) \arrow[r, "\lambda"'] & \Omega_S(\mathcal{L}) \arrow[u, "\kappa"]
\end{tikzcd} \tag{13}$$

where μ'_f is the map that assigns to each partially connective criterion σ the partial connection $\{c \in \mathcal{L}; \sigma(f,c) = 1\}$, and κ is the map that assigns to each class splitting opening γ^* the partial S-partition $\gamma^*(1)$ (where 1 *is the greatest S-partition on $\mathcal{L}$).*

The proof is analogous to that of Theorem 4. Theorems 4 and 5 thus relate the two approaches to connective segmentation discussed in Sections 1–3.

5 Conclusion

As we have seen, the theory of connective segmentation generalizes to atomistic lattices in a natural way. When dealing with non-atomistic lattices, partiality becomes necessary in order to maintain the full strength of the theory. We have also seen that the two approaches to connective segmentation are linked on all levels. In short, the apparatus of connective segmentation lends itself to investigations of a large number of complete lattices that arise naturally in different theoretical and practical contexts of mathematical morphology.

References

1. Braga-Neto, U., Goutsias, J.: Connectivity on Complete Lattices: New Results. Comput. Vis. Image. Underst. 85, 22–53 (2001)
2. Ore, O.: Galois connexions. Trans. Amer. Math. Soc. 55, 493–513 (1944)
3. Ronse, C.: Partial Partitions, Partial Connections and Connective Segmentation. J. Math. Imaging Vis. 32, 97–125 (2008)
4. Ronse, C.: Idempotent Block Splitting on Partial Partitions, I: Isotone Operators. Order. SpringerLink (2010)
5. Ronse, C., Serra, J.: Geodesy and Connectivity in Lattices. Fundamenta Informaticae 46, 349–395 (2001)
6. Ronse, C., Serra, J.: Algebraic foundations of morphology. In: Najman, L., Talbot, H. (eds.) Mathematical Morphology: From Theory to Applications. ISTE/J. Wiley & Sons, London (2010)
7. Serra, J.: Viscous Lattices. J. Math. Imaging Vis. 22, 269–282 (2005)
8. Serra, J.: A Lattice Approach to Image Segmentation. J. Math. Imaging Vis. 24, 83–130 (2006)

Hyperconnections and Openings on Complete Lattices

Michael H.F. Wilkinson

Johann Bernoulli Institute of Mathematics and Computer Science,
University of Groningen, P.O. Box 407, 9700 AK Groningen, The Netherlands
m.h.f.wilkinson@rug.nl

Abstract. In this paper the notion of hyperconnectivity, which is an extension of connectivity is explored in the lattice theoretical framework. It is shown that a fourth axiom is needed when moving from connections to hyperconnections, in order to define hyperconnected components meaningfully, which is important for the definition of, e.g., viscous levellings. New hyperconnectivity openings, which are the hyperconnected equivalents of connectivity openings are then defined. It then shown that *all* algebraic openings which are translation and grey-scale invariant can be described as hyperconnected attribute filters. This means that hyperconnectivity lies at the heart of a vast range of morphological filters.

1 Introduction

Hyperconnectivity was first put forward by Serra [8] as a generalization of connectivity. As discussed in [10, 11], hyperconnectivity offers the prospect of generalizing a large number of morphological filters into a single framework, and the existence of a continuum of filters with properties intermediate between connected and structural filters was proposed. Fig. 1 shows an example comparing structural open-close with connected and viscous hyperconnected levellings. Though viscous *reconstruction* can readily be described in terms of a connection on a viscous lattice [9], this is awkward for a levelling, because the viscous lattice, consisting of the set of dilates of members of some lattice $\mathcal{L}$, is not complemented. This means we cannot define auto-dual filters because the complement of a member of a viscous lattice is not necessarily a member of that lattice. Because viscous hyperconnections are defined in terms of the original lattice in [10], we can define auto-dual filters such as levellings without difficulty. This is one reason why we need to define hyperconnections for complete lattices.

A shortcoming of [10], was that only the finite, set-theoretical case was discussed. In this paper, the theory is extended to the compete lattice framework, and the infinite case. The new theory provides deeper understanding of hyperconnectivity, and provides a theoretical basis for grey-scale hyperconnected filters such as those developed in [5, 4]. I will first define some basic notions and notation, and then recall the theory of connectivity classes or connections, as developed by Serra in the complete lattice case [8]. In the same paper Serra put forward the notion of hyperconnections, by generalizing the third axiom for connections. I will show that this generalization is insufficient to define hyperconnected components properly, and that a fourth axiom is required. This extra axiom allows definition of new hyperconnectivity openings, which generalize the operators developed in the finite, set theoretical case in [10]. In that paper, and in [11],

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 73–84, 2011.

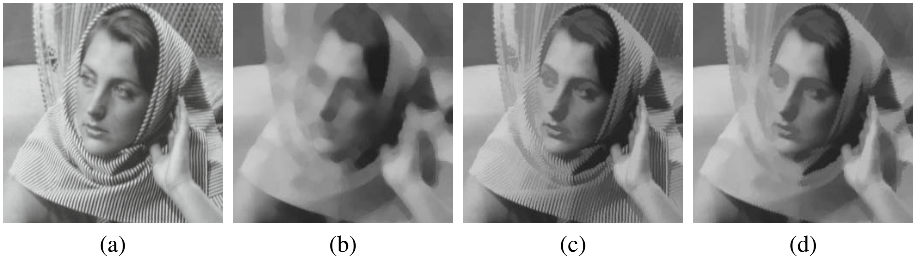


Fig. 1. Self-dual filters: (a) original image f; (b) point-wise average of structural open-close, and close-open of f; (c) levelling of f using (b); (c) equivalent viscous hyperconnected levelling

it was shown that structural openings, viscous filters, and path openings all were special cases of hyperconnected filters. In this paper I will show that all openings which when applied to any canonical sup-generator (e.g., impulse function) returns the zero element of the lattice is a hyperconnected attribute filter. This means that hyperconnectivity lies at the heart of a vast number of morphological filters.

2 Theory

In the following $\mathcal{L}$ denotes a complete lattice, with global infimum and supremum $\mathbf{0}$ and $\mathbf{1}$, and $\mathcal{P}(\mathcal{L})$ the power lattice of $\mathcal{L}$. Elements of $\mathcal{L}$ are denoted as lower-case a, b, c; elements of $\mathcal{P}(\mathcal{L})$ as upper-case A, B, C, etc. An algebraic opening is any operator γ which is idempotent ($\gamma(\gamma(a)) = \gamma(a)$), increasing ($a \leq b \Rightarrow \gamma(a) \leq \gamma(b)$) and anti-extensive ($a \leq \gamma(a)$). A cover C of any $a \in \mathcal{L}$ is an element of $\mathcal{P}(\mathcal{L})$ such that $\bigvee C = a$. A partition $P \in \mathcal{P}(\mathcal{L})$ of any $a \in \mathcal{L}$, is a cover such that any two $x_1, x_2 \in P$ are either equal or disjoint. A chain is any totally ordered set.

2.1 Redundancy Operators

Any non-empty $A \in \mathcal{P}(\mathcal{L})$ is the union of a set of maximal chains C_i, because any chain $C \subseteq A$ can be extended to a maximal chain according to the Hausdorff maximal principle. For every element $a \in A$, $\{a\}$ is a chain, which can be extended to a maximal chain C_i with $\{a\} \subseteq C_i \subseteq A$. Thus the union $\bigcup_i C_i \subseteq A$, because all maximal chains $C_i \subseteq A$. However, because all $a \in A$ are contained in at least one chain C_i, $A \subseteq \bigcup_i C_i$, and thus $A = \bigcup_i C_i$. This leads to the definition of *non-redundancy*:

Definition 1 (Non-redundancy). *Any element $A \in \mathcal{P}(\mathcal{L})$ is said to be* non-redundant *if*

$$a_i \leq a_j \quad \Rightarrow \quad a_i = a_j, \quad \forall a_i, a_j \in A \tag{1}$$

or, equivalently, that all maximal chains $C_i \subseteq A$ have cardinality $\#(C_i) = 1$.

It is readily seen that all partitions are non-redundant covers, but not all non-redundant covers are partitions.

Any set $A \in \mathcal{P}(\mathcal{L})$ is said to be *chain-sup-complete* if for every non-empty chain $C \subseteq A$ we have $\bigvee C \in A$. All non-redundant sets, and all finite A are chain-sup-complete. Note that if the supremum of the empty chain ($= \mathbf{0}$) is also member of A it is *chain complete* or *ω-complete* as in [3]. It can readily be shown that the union of two chain-sup-complete sets is itself chain-sup-complete. The set of all chain-sup-complete elements of $\mathcal{P}(\mathcal{L})$ is denoted as $\mathcal{P}_{csc}(\mathcal{L})$.

We define $\mathcal{N}(\mathcal{L}) \subseteq \mathcal{P}_{csc}(\mathcal{L}) \cup \{\emptyset\}$ as the family of all non-redundant elements of $\mathcal{P}_{csc}(\mathcal{L}) \cup \{\emptyset\}$. On $\mathcal{N}(\mathcal{L})$ we define the *refinement order* $\sqsubseteq$ as used in partitions

$$A \sqsubseteq B \quad \equiv \quad \forall a_i \in A \ \ \exists b_j \in B : a_i \leq b_j \tag{2}$$

If $A_1 \sqsubseteq A_2$ for two partitions or covers we state that A_1 is finer than A_2, or, equivalently, A_2 is coarser than A_1. Relation $\sqsubseteq$ is a partial order $\mathcal{N}(\mathcal{L})$, and a partial preorder on $\mathcal{P}(\mathcal{L})$, i.e., it is reflexive and transitive, but not antisymmetric. We can now define the *reduction operator* which is crucial to the selection of hyperconnected components:

Definition 2 (Reduction Operator). *The reduction operator $\psi_{\mathcal{N}} : \mathcal{P}(\mathcal{L}) \to \mathcal{N}(\mathcal{L})$ is defined as*

$$\psi_{\mathcal{N}}(A) = \{a \in A \mid \nexists b \in A : a < b\} \tag{3}$$

Because $\psi_{\mathcal{N}}(A) \subseteq A$, we can readily see from the definition that no two elements $a, b \in \psi_{\mathcal{N}}(A)$ exist such that $a \leq b$, and $\psi_{\mathcal{N}}(A) \in \mathcal{N}(\mathcal{L})$.

Proposition 1. *For any set $A \in \mathcal{P}_{csc}(\mathcal{L})$*

1. *$\psi_{\mathcal{N}}(A)$ is the least element in $\mathcal{N}(\mathcal{L})$ with $A \sqsubseteq \psi_{\mathcal{N}}(A)$*
2. *$\bigvee A = \bigvee \psi_{\mathcal{N}}(A)$*

Proof. If $A \in \mathcal{P}_{csc}(\mathcal{L})$, $\psi_{\mathcal{N}}(A)$ contains all maximal elements of A, and nothing else. Therefore, $A \sqsubseteq \psi_{\mathcal{N}}(A)$. If we obtain some $B \in \mathcal{N}(\mathcal{L})$, with $B \sqsubset \psi_{\mathcal{N}}(A)$, by either removing any element $a \in \psi_{\mathcal{N}}(A)$, or replacing it by some $b < a$, this means there exists no $c \in B$ such that $a \leq c$. Because $a \in A$ we have $A \not\sqsubseteq B$, proving property 1.

To prove the second point, we can write

$$A = \bigcup_i \{C_i\} \quad \Rightarrow \quad \bigvee A = \bigvee_i \left\{ \bigvee C_i \right\} \tag{4}$$

with $\{C_i\}$ the set of all maximal chains $C_i \subseteq A$. Because A is chain sup-complete, it contains the maximal elements of all its non-empty chains C_i, and we may write

$$\psi_{\mathcal{N}}(A) = \left\{ \bigvee C_i \right\} \quad \Rightarrow \quad \bigvee \psi_{\mathcal{N}}(A) = \bigvee_i \left\{ \bigvee C_i \right\} = \bigvee A \tag{5}$$

which proves Proposition 1.

If $A \notin \mathcal{P}_{csc}(\mathcal{L})$, these properties do not necessarily hold. For example, let A be the set of open balls b_r of radius r defined as $A = \{b_{(1-1/n)}, n \in \mathbb{Z}^+\}$. The supremum of this set is the open ball $b_1 \notin A$, and thus $A \notin \mathcal{P}_{csc}(\mathcal{L})$. Because for any element $a \in A$ there exist infinitely many $b \in A$ such that $a < b$, we obtain $\psi_{\mathcal{N}}(A) = \emptyset$. This violates both properties in Proposition 1.

2.2 Connectivity

Connectivity such as is used in morphological filtering is defined through the notion of connectivity classes [8,6] defined as follows.

Definition 3. *A connectivity class $\mathcal{C} \subseteq \mathcal{P}(\mathcal{L})$ is a set of elements of $\mathcal{L}$ with the following three properties:*

1. $\mathbf{0} \in \mathcal{C}$
2. *$\mathcal{C}$ is sup-generating,*
3. *for each family $\{c_i\} \subseteq \mathcal{C}$, $\bigwedge c_i \neq \mathbf{0}$ implies $\bigvee c_i \in \mathcal{C}$.*

Any element $c \in \mathcal{C}$ is said to be connected. Any element $a \in \mathcal{L}$ can be partitioned into connected components. These are the elements of $c \in \mathcal{C}$ such that $c \leq a$ of maximal extent, i.e., if $c \leq a$, $c \in \mathcal{C}$, and there exists no $d \in \mathcal{C}$ such that $c < d \leq a$, then c is a connected component of a. Let $\mathcal{C}_a$ be defined as follows:

$$\mathcal{C}_a = \{c \in \mathcal{C} \mid \mathbf{0} < c \leq a\}, \tag{6}$$

in other words $\mathcal{C}_a$ is the set of all elements of $\mathcal{C} \setminus \{\mathbf{0}\}$ majorated by a. The set of all connected components $\mathcal{C}_a^*$ of a is simply

$$\mathcal{C}_a^* = \psi_{\mathcal{N}}(\mathcal{C}_a). \tag{7}$$

We can do this because of the following property.

Proposition 2 (Chain completeness and connections). *Any connection $\mathcal{C}$ on any complete lattice $\mathcal{L}$ is chain complete.*

Proof. The supremum of the empty chain is $\mathbf{0} \in \mathcal{C}$. Let $B \subseteq \mathcal{C}$ be a non-empty chain. If $\bigwedge B \neq \mathbf{0}$, then $\bigvee B \in \mathcal{C}$ by definition. If $\bigwedge B = \mathbf{0}$, and $\bigvee B \neq \mathbf{0}$ we pick an arbitrary $m \in B$ such that $\mathbf{0} < m \leq \bigvee B$ and define the set

$$B^+ = \{b_i \in B \mid m \leq b_i\}. \tag{8}$$

Thus, $\mathbf{0} < m \leq \bigwedge B^+$ and therefore $\bigvee B^+ \in \mathcal{C}$. Obviously

$$\bigvee B = \bigvee B^+ \quad \Rightarrow \quad \bigvee B \in \mathcal{C}. \tag{9}$$

Finally, if $\bigvee B = \mathbf{0}$ we have $\bigvee B \in \mathcal{C}$ as well, proving Proposition 2.

The connected components of any image can be accessed through *connectivity openings* [8,1]:

Definition 4. *The connectivity opening γ_x of $a \in \mathcal{L}$ marked by some $x \in \mathcal{C} \setminus \{\mathbf{0}\}$, is given by*

$$\gamma_x(a) = \begin{cases} \bigvee\{c_i \in \mathcal{C} \mid x \leq c_i \leq a\} & \text{if } x \leq a \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{10}$$

In this definition the notion of maximum extent is derived by taking the supremum of all connected subsets of a but larger than or equal to x. It can readily be shown that this is equivalent to

$$\gamma_x(a) = \begin{cases} c_i \in \mathcal{C}_a^* : x \leq c_i & \text{if } x \leq a \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{11}$$

This equivalence stems from the fact that c_i in (10) for which of $x \leq c_i \leq a$ have infimum $x \neq \mathbf{0}$, and that their supremum is therefore connected. Serra [8] notes that in many important cases it is sufficient to use some canonical set of sup-generators $\mathcal{S}$ to obtain all connected components of any $a \in \mathcal{L}$, through a family of connectivity openings $\{\gamma_s, s \in \mathcal{S}\}$. Proposition 2 shows that if the smallest connection possible of $\mathcal{L}$ is $\{\mathbf{0}\} \cup \mathcal{S}$, as asserted in [8,6], this is only true if $\mathcal{S}$ is chain-sup-complete.

2.3 Hyperconnectivity

Hyperconnectivity extends connectivity by generalizing the third condition of Definition 3 [8]. Instead of using a non-empty intersection, we can use any *overlap criterion* $\perp$ which is defined as follows.

Definition 5. *An overlap criterion in $\mathcal{L}$ is a mapping $\perp : \mathcal{P}(\mathcal{L}) \to \{$false, true$\}$ such that $\perp$ is decreasing, i.e., for any $A, B \in \mathcal{P}(\mathcal{L})$*

$$A \subseteq B \quad \Rightarrow \quad \perp(B) \leq \perp(A). \tag{12}$$

Note that *false* $\leq$ *true*. Any $A \in \mathcal{P}(\mathcal{L})$ for which $\perp(A) =$ *true* is said to be *overlapping*. We can now define a *hyperconnectivity class* as follows.

Definition 6. *A hyperconnectivity class $\mathcal{H} \subseteq \mathcal{L}$ is a class with the following properties:*

1. $\mathbf{0} \in \mathcal{H}$
2. *$\mathcal{H}$ is sup-generating*
3. *$\mathcal{H}$ is chain-sup complete,*
4. *for each family $\{h_i\} \subset \mathcal{H}$, $\perp(\{h_i\})$ implies $\bigvee_i h_i \in \mathcal{H}$,*

with $\perp$ an overlap criterion, for which $\perp(A) \Rightarrow \bigwedge A \neq \mathbf{0}$.

The above definition of hyperconnections differs from that given by Serra [8], in that it has four axioms, rather than just three. The additional axiom (number 3) is necessary in the infinite case, as will be discussed shortly. If a canonical family of sup-generators $\mathcal{S}$ exists, axiom 2 can be rewritten to $s \in \mathcal{H}$ for all $s \in \mathcal{S}$ as in the connected case [8,6].

Serra [8] showed that connectivity is an extension of hyperconnectivity, in which the third axiom in Def. 3 is replaced by a stricter requirement. For example we might require that there exist a ball b_r of some diameter r for which $b_r \leq \bigwedge_i h_i$, leading to viscous hyperconnections [10]. Axioms 1 and 3 of Def. 6 mean hyperconnections are chain complete, just like connections.

Like the notion of *connected* components for connectivity classes, we need to define the notion of *hyperconnected components* of $a \in \mathcal{L}$, which are members of the hyperconnection $h_i \in \mathcal{H}$, such that $h_i \leq a$ has maximal extent. In complete analogy with connected components we can first define the set $\mathcal{H}_a$ of all elements of $\mathcal{H}$ such that $\mathbf{0} < h_i \leq a$:

$$\mathcal{H}_a = \{h \in \mathcal{H} \mid \mathbf{0} < h \leq a\}. \tag{13}$$

$\mathcal{H}_a$ is either chain-sup-complete or empty. The latter is the case if $a = \mathbf{0}$. If $a \neq \mathbf{0}$, there exists at least one $h \in \mathcal{H}$ such $\mathbf{0} < h \leq a$, because $\mathcal{H}$ is sup-generating. In that case, consider any non-empty chain $\{h_i\} \subset \mathcal{H}_a$. Obviously,

$$\bigvee_i h_i \in \mathcal{H}, \tag{14}$$

through axiom 3 in Def. 6. Because all $h_i \leq a$ it follows that

$$\bigvee_i h_i \leq a, \quad \Rightarrow \quad \bigvee_i h_i \in \mathcal{H}_a. \tag{15}$$

This proves that $\mathcal{H}_a$ is chain-sup-complete if $a \neq \mathbf{0}$. The set of hyperconnected components $\mathcal{H}_a^*$ of a is defined equivalently

$$\mathcal{H}_a^* = \psi_{\mathcal{N}}(\mathcal{H}_a). \tag{16}$$

Unlike $\mathcal{C}_a^*$, $\mathcal{H}_a^*$ is not necessarily a partition of a, because two hyperconnected components h_j, h_k, with $h_j \neq h_k$, may have a non-zero infimum, but $h_j \vee h_k$ is not a member of $\mathcal{H}_a$ because $\perp(\{h_j, h_k\}) = $ *false*. This is easily understood, because if $\perp(\{h_j, h_k\}) = $ *true*, then $h_j \vee h_k \in \mathcal{H}_a$. However, because $h_j, h_k \in \mathcal{H}_a^*$, we have $h_j \not\leq h_k$ and $h_k \not\leq h_j$ because of the definition of $\psi_{\mathcal{N}}$. Thus $h_j < h_j \vee h_k$ and $h_k < h_j \vee h_k$. This means $h_j \vee h_k$ is a larger element of $\mathcal{H}_a$ than either h_j or h_k which contradicts their membership of $\mathcal{H}_a^*$. Note that due to Prop. 1, $\mathcal{H}_a^*$ only is a cover of a if $\mathcal{H}$ is chain-sup-complete. This is why we need the third axiom in Def. 6 in the infinite case.

2.4 Hyperconnectivity Openings

We now introduce the families of hyperconnectivity openings $\Upsilon_h : \mathcal{N}(\mathcal{L}) \to \mathcal{N}(\mathcal{L})$, with $h \in \mathcal{H}$, which return sets of hyperconnected components. If the lattice is supplied with a canonical set of sup-generators $\mathcal{S}$, we may restrict the family $\{\Upsilon_h, h \in \mathcal{H}\}$ to $\{\Upsilon_s, s \in \mathcal{S}\}$, as before. As in [10] we need the reduction operator.

Definition 7. *The hyperconnectivity opening $\Upsilon_x : \mathcal{N}(\mathcal{L}) \to \mathcal{N}(\mathcal{L})$, with $x \in \mathcal{H} \setminus \{\mathbf{0}\}$, associated with hyperconnectivity class $\mathcal{H}$ is defined as*

$$\Upsilon_x(A) = \psi_{\mathcal{N}}\left(\bigcup_{a \in A} \{h \in \mathcal{H}_a \mid x \leq h\}\right) \tag{17}$$

If the parameter A is just a singleton $\{a\}$, Υ_x extracts the set of hyperconnected components of a containing x. Note that $\Upsilon_x(\emptyset) = \Upsilon_x(\{\mathbf{0}\}) = \emptyset$, and more generally that if there exists no $a \in A$ such that $x \leq a$ we have $\Upsilon_x(A) = \emptyset$.

Theorem 1 **($\varUpsilon_x$ are algebraic openings).** *For any hyperconnection $\mathcal{H}$ on any complete lattice $\mathcal{L}$, $\{\varUpsilon_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}\}$ is a family of algebraic openings on $(\mathcal{N}(\mathcal{L}), \sqsubseteq)$, for which*

$$\{\mathbf{0}\} \cup \left(\bigcup_{x \in \mathcal{H} \setminus \{\mathbf{0}\}} \bigcup_{A \in \mathrm{Inv}(\varUpsilon_x)} A \right) = \mathcal{H}. \tag{18}$$

$\mathrm{Inv}(\gamma) \subseteq \mathcal{L}$ denotes the *invariance domain* of γ, which is the set $\{a \in \mathcal{L} \mid a = \gamma(a)\}$.

Proof. By definition, the output of $\varUpsilon_h(A)$ is a non-redundant subset of $\mathcal{H}$ such that for each element $h_i \in \varUpsilon_x(A) \subseteq \mathcal{H}$ we have $s \leq h_i$. Furthermore, $\varUpsilon_x(A) \sqsubseteq A$, because we can see from the definition that for every $h_i \in \varUpsilon_x(A)$ there exists an $a \in A$ such that $h_i \leq a$, which proves anti-extensiveness in terms of $\sqsubseteq$ as defined in (2).

It is readily verified that $\varUpsilon_x(\emptyset) = \emptyset$ for all $x \in \mathcal{H} \setminus \{\mathbf{0}\}$, so idempotence holds whenever $\varUpsilon_x(A) = \emptyset$. Let $\{h_i\} = \varUpsilon_x(A) \neq \emptyset$, with i from some index set. Applying $\varUpsilon_x$ to $\{h_i\}$ is equivalent to applying the reduction operator to the union of all sets

$$\mathcal{H}^s_{h_i} = \{h_j \in \mathcal{H} \mid s \leq h_j \leq h_i\}. \tag{19}$$

All $\mathcal{H}^s_{h_i}$ contain h_i because $s \leq h_i$ and $h_i \in \mathcal{H}$. Thus each $\mathcal{H}^s_{h_i}$ contains h_i as maximal element. Therefore we have

$$\varUpsilon_x(\varUpsilon_x(A)) = \varUpsilon_x(\{h_i\}) = \psi_{\mathcal{N}}\left(\bigcup_i \mathcal{H}^s_{h_i} \right) = \{h_i\} \tag{20}$$

because the reduction operator removes all elements h_j of every $\mathcal{H}^s_{h_i}$, such that $h_j < h_i$, for each $h_i \in \varUpsilon_x(A)$. All elements $h_i \in \varUpsilon_x(A)$ are preserved because it is a non-redundant set, which cannot be reduced any further. Thus

$$\varUpsilon_x(\varUpsilon_x(A)) = \varUpsilon_x(A) \tag{21}$$

and $\varUpsilon_x(A)$ is idempotent. The above arguments also show that all elements of the invariance domain of any $\varUpsilon_x$ consist of subsets of $\mathcal{H}$. Furthermore, any non-zero $h_i \in \mathcal{H}$ is marked by at least one sup-generator, so that every non-zero element of $\mathcal{H}$ is contained in the union of the elements of the invariance domains of all $\varUpsilon_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}$, simply because $\varUpsilon_x(\{h_i\}) = \{h_i\}$, for any $x \leq h_i$. Thus, the union of all elements of the invariance domains of all $\varUpsilon_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}$, augmented with $\mathbf{0}$ is equal to $\mathcal{H}$.

Finally, we must show that $\varUpsilon_x$ is increasing in terms of $\sqsubseteq$. Let $A, B \in \mathcal{N}(\mathcal{L})$ with $A \sqsubseteq B$. This means that for every $a \in A$ there exists a $b \in B$ such that $a \leq b$. We already know that for every $h_i \in \varUpsilon_x(A)$ there exists an $a \in A$ such that $h_i \leq a$. Therefore, there exists a $b \in B$ such that $h_i \leq b$. This means that either $h_i \in \varUpsilon_x(B)$, or there exists an $h_j \in \varUpsilon_x(B)$ such that $h_i < h_j$. Thus, for every $h_i \in \varUpsilon_x(A)$ there exists an $h_j \in \varUpsilon_x(B)$ such that $h_i \leq h_j$, and thus

$$A \sqsubseteq B \quad \Rightarrow \quad \varUpsilon_x(A) \sqsubseteq \varUpsilon_x(B), \tag{22}$$

proving increasingness, and all $\varUpsilon_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}$ are algebraic openings, proving Theorem 1, and that the family $\{\varUpsilon_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}\}$ retrieves $\mathcal{H}$, through (18).

What needs to be done is to assess which properties a family of mappings $\Upsilon_x : \mathcal{N}(\mathcal{L}) \to \mathcal{N}(\mathcal{L})$ requires for it to define a hyperconnectivity class, in the same way a family of connectivity openings defines a connectivity.

Theorem 2 (Hyperconnectivity Openings). *On any lattice $\mathcal{L}$, every hyperconnection $\mathcal{H}$ associated to an overlap criterion $\perp$ is equivalent to a family of algebraic openings $\{\Upsilon_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}\}$ on $(\mathcal{N}(\mathcal{L}), \sqsubseteq)$ with the following properties*

1. *Υ_x is an algebraic opening indexed by $x \in \mathcal{H} \setminus \{\mathbf{0}\}$.*
2. *for all $x \in \mathcal{H} \setminus \{\mathbf{0}\}$ we have $\Upsilon_x(\{x\}) = \{x\}$*
3. *for all $A \in \mathcal{N}(\mathcal{L})$, and all $x \in \mathcal{H} \setminus \{\mathbf{0}\}$ we have $s \not\leq a, \forall a \in A \ \Rightarrow \ \Upsilon_x(A) = \emptyset$;*
4. *for any $x, y \in \mathcal{H} \setminus \{\mathbf{0}\}$ and any $A \in \mathcal{P}(\mathcal{L})$, $h_i \in \Upsilon_x(A)$ and $y \leq h_i \ \Rightarrow \ h_i \in \Upsilon_y(A)$.*
5. *for all $x \in \mathcal{H} \setminus \{\mathbf{0}\}$ and all $a \in \mathcal{L}$, and any $\{h_i\} \subseteq \Upsilon_x(\{a\})$ we have $\#\{h_i\} \neq 1 \ \Rightarrow \ \perp(\{h_i\}) = false$.*

Proof. The first property follows directly from Theorem 1, and its proof. The second property follows directly from Def. 7 and the fact that

The third property derives directly from Def. 7, because (17) implies that if $x \not\leq a$ for all $a \in A$, $\Upsilon_x(A) = \emptyset$. Together with the first two this guarantees $\bigvee \Upsilon(A) = \bigvee A$.

The fourth property can be derived directly from (17): Because $h_i \in \Upsilon_x(A)$, there exists no hyperconnected set $h_j \in \Upsilon_x(A)$, such that $h_i < h_j$, because $\Upsilon_x(A)$ is non-redundant. If $t \leq h_i$ but $h_i \notin \Upsilon_t(A)$, this would imply that there is some $h_j \in \Upsilon_x(A)$, such that $h_i \leq h_j$, leading to contradiction.

The fifth property states that no set of two or more sets $\{h_i\} \in \Upsilon_x(\{a\})$ can overlap in the sense of $\perp$. If they did, $\bigvee_i h_i \in \mathcal{H}$ and $s \leq \bigvee_i h_i \in \Upsilon_x(\{a\}$, and $\{h_i\} \not\subset \Upsilon_x(\{a\})$.

We now show that any family of operators $\{\Upsilon_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}\}$ with the above properties for a given overlap criterion $\perp$ is associated with a hyperconnection. Suppose we have some family $\{\Upsilon_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}\}$ of algebraic openings marked by $x \in \mathcal{H} \setminus \{\mathbf{0}\}$. Let $\mathcal{I}$ be defined as

$$\mathcal{I} = \{\mathbf{0}\} \cup \left(\bigcup_{x \in \mathcal{H}} \ \bigcup_{A \in \mathrm{Inv}(\Upsilon_x)} A \right), \tag{23}$$

for our family $\{\Upsilon_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}\}$. We must now show that $\mathcal{I}$ is a hyperconnection.

The second property implies $\mathcal{H} \subset \mathcal{I}$, proving $\mathcal{I}$, is sup-generating, conforming to property 2 of Def. 6. The definition of $\mathcal{I}$ states that $\mathbf{0} \in \mathcal{I}$, meeting property 1 of Def. 6. Furthermore, let $C \subset \mathcal{I}$ be a chain. This implies $\Upsilon_x(\{\bigvee C\}) = \{\bigvee C\}$. If not, we would have to represent the hyperconnected components $\bigvee C$ as a chain with cardinality larger than one, which contradicts the fact that $\Upsilon_x(\{\bigvee C\}) \in \mathcal{N}(\mathcal{L})$. Thus $\mathcal{I}$ is chain sup complete.

Similarly, let $\perp(H) =$ *true* for some $H \subseteq \mathcal{I}$. According to property 5, and the fact that $\bigvee \Upsilon(A) = \bigvee A$ for any $A \in \mathcal{N}(\mathcal{L})$, we have $\Upsilon_x(\{\bigvee H\}) = \bigvee H$ and thus $\perp(H) =$ *true* guarantees $\bigvee H \in \mathcal{I}$, and $\mathcal{I}$ is a hyperconnection.

Finally, we show that applying (17) to the hyperconnection $\mathcal{I}$, we find the same family of openings we started out with, note that $\mathcal{H} \subseteq \mathcal{I}$, due to property 2 in Theorem 2.

Let the family $\{\Upsilon^{\mathcal{I}}_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}\}$ be the new family of operators, with each $\Upsilon^{\mathcal{I}}_x$ defined as

$$\Upsilon^{\mathcal{I}}_x(A) = \psi_{\mathcal{N}}\left(\bigcup_{a \in A} \{h \in \mathcal{I}_a \mid x \leq h\}\right) \tag{24}$$

which is just rewriting (17), by replacing $\mathcal{H}$ with $\mathcal{I}$. It is readily verified that the invariance domain $\mathrm{Inv}(\Upsilon^{\mathcal{I}}_x)$ of each $\Upsilon^{\mathcal{I}}_x$ is given by

$$\mathrm{Inv}(\Upsilon^{\mathcal{I}}_x) = \{A \in \mathcal{N}(\mathcal{I}) \mid \forall a \in A : x \leq a\} \tag{25}$$

with $\mathcal{N}(\mathcal{I})$ the family of non-redundant subsets of $\mathcal{I}$. This is true because the input of $\Upsilon^{\mathcal{I}}_x$ must be non-redundant by definition, and if and only if the input A of $\Upsilon^{\mathcal{I}}_x$ is a subset of $\mathcal{I}$ and all elements of A are larger than or equal to x is A left unaffected by $\Upsilon^{\mathcal{I}}_x$. For the original family $\{\Upsilon_x, x \in \mathcal{H} \setminus \{\mathbf{0}\}\}$, we have that all members of $\mathrm{Inv}(\Upsilon_x)$, must also be non-redundant, and a subset of $\mathcal{I}$, each element of which must be larger than or equal to x, due to properties 2 and 3 of Theorem 2, and increasingness of openings. Thus,

$$\mathrm{Inv}(\Upsilon_x) = \{A \in \mathcal{N}(\mathcal{I}) \mid \forall a \in A : x \leq a\} = \mathrm{Inv}(\Upsilon^{\mathcal{I}}_x) \tag{26}$$

for all $x \in \mathcal{H} \setminus \{\mathbf{0}\}$. Because any opening γ maps the input A to the largest element $B \in \mathrm{Inv}(\gamma)$ such that $B \leq A$, the invariance domain uniquely defines an opening. This means $\Upsilon_x = \Upsilon^{\mathcal{I}}_x$ for all $x \in \mathcal{H} \setminus \{\mathbf{0}\}$, proving Theorem 2.

3 Openings and Hyperconnections

We now turn to the question of which openings form part of the gamut of hyperconnected filters. In this discussion we restrict ourselves to the case in which a *chain-sup-complete*, canonical set $\mathcal{S}$ of sup-generators exists, i.e. $\mathcal{S} \cup \{\mathbf{0}\}$ is a minimal connection. In the binary case we have $\mathcal{S} = \{\{x\} \mid x \in E\}$. In the lattice of functions from E to $\mathbb{R} \cup \{-\infty, +\infty\}$ we obtain $\mathcal{S} = \{\delta^a_x, x \in E, a \in \mathbb{R} \cup \{+\infty\}\}$, with

$$\delta^a_x(y) = \begin{cases} a & \text{if } x = y \\ -\infty & \text{otherwise,} \end{cases} \tag{27}$$

i.e., we need to include both finite and infinite impulses.

In the following we need the property that $\mathrm{Inv}(\gamma)$ is closed under supremum [7] for any algebraic opening. In analogy to connected filters we arrive at the following definition.

Definition 8 (Hyperconnected openings). *An opening $\gamma : \mathcal{L} \to \mathcal{L}$ is* hyperconnected *if there exists a hyperconnection $\mathcal{H}$ such that*

$$\mathcal{H}^*(\gamma(a)) \subseteq \mathcal{H}^*(a) \quad \forall a \in \mathcal{L} \tag{28}$$

The meaning of this is that a hyperconnected opening only removes hyperconnected components. Due their anti-extensive nature, no opening can extend an existing hyperconnected component. In analogy with connected filters, no new components may arise. Thus, any hyperconnected component present in the opening must be present in the original.

Hyperconnected attribute filters can be defined in much the same way as connected attribute filters, using trivial filters. A *trivial filter* $\Psi_\Lambda(h)$ based on criterion $\Lambda : \mathcal{H} \rightarrow \{\mathit{false}, \mathit{true}\}$ returns h if $\Lambda(h) = 1$, and $\mathbf{0}$ otherwise, for any $h \in \mathcal{H}$. Let $\Psi_\Lambda(\mathcal{H}_a^*)$ be shorthand for the set of all $h_j \in \mathcal{H}_a^*$ such that $\Lambda(h_j) = 1$. A hyperconnected attribute filter $\Psi^\Lambda : \mathcal{P}(\mathcal{L}) \rightarrow \mathcal{P}(\mathcal{L})$ based on criterion $\Lambda : \mathcal{H} \rightarrow \{\mathit{false}, \mathit{true}\}$ is defined as

$$\Psi^\Lambda(a) = \bigvee_{s \leq X} \bigvee_{h_i \in \Upsilon_x(a)} \Psi_\Lambda(h_i) = \bigvee_{h_j \in \mathcal{H}_a^*} \Psi_\Lambda(h_j) = \bigvee_{h_k \in \Psi_\Lambda(\mathcal{H}_a^*)} h_k, \tag{29}$$

The following theorem links openings with hyperconnections.

Theorem 3 (Hyperconnections and openings). *Every algebraic opening γ on any complete, sup-generated lattice $\mathcal{L}$ with a* chain-sup-complete *set of sup-generators $\mathcal{S}$ is associated with a hyperconnection $\mathcal{H}_\gamma$ given by*

$$\mathcal{H}_\gamma = \mathrm{Inv}(\gamma) \cup \mathcal{S}. \tag{30}$$

with overlap criterion

$$\perp_\gamma(A) = \bigwedge A \neq \mathbf{0} \quad \wedge \quad A \subseteq \mathrm{Inv}(\gamma) \tag{31}$$

Proof. Because $\mathbf{0} = \gamma(\mathbf{0})$ by anti-extensiveness, and inclusion of the sup-generators, we adhere to the first two properties of hyperconnections. It is trivial to show that $\perp_\gamma$ is decreasing because both terms are decreasing, and all must be met. The first term of (31) guarantees that the infimum of A is not $\mathbf{0}$, which meets the requirement that $\perp_\gamma$ must imply a non-zero infimum. The second term states that $A \subseteq \mathrm{Inv}(\gamma)$, which guarantees $\bigvee A \in \mathcal{H}_\gamma$, because $\mathrm{Inv}(\gamma)$ is closed under supremum.

Finally, $\mathcal{H}_\gamma$ is the union of of two chain-sup-complete sets, and is therefore chain-sup-complete. Therefore $\mathcal{H}_\gamma$ is a hyperconnection, proving Theorem 3.

We have now shown any opening on any complete lattice, sup-generated by a chain-sup-complete family $\mathcal{S}$ to be associated with a hyperconnection, but that does not prove they are hyperconnected openings. For that we need to see whether the result of any opening contains only hyperconnected components of the input.

The hyperconnected components of any $a \in \mathcal{L}$ are simply $\gamma(a)$ augmented with such sup-generators as are required to "fill the gaps" between a and $\gamma(a)$, or

$$\mathcal{H}_\gamma^*(a) = \psi_\mathcal{N}(\{\gamma(a)\} \cup \{s \in \mathcal{S} | s \leq a\}). \tag{32}$$

This yields a non-redundant cover of a because $\mathcal{H}_\gamma$ is a hyperconnection if $\mathcal{S}$ is chain-sup complete. For example, take the image in Fig. 2(a). The main hyperconnected component $\gamma(f)$ is shown in Fig. 2(b). The remaining hyperconnected components are impulse functions at all points (x, y) where $\gamma(f)(x, y) \neq f(x, y)$. The height of each impulse function at (x, y) is given by $f(x, y)$, and their supremum is shown in Fig. 2(c). Fig. 2(d) shows that the infimum of (b) and(c) is not zero. We now consider the following opening γ_s with $s \in \mathcal{S}$, defined as

$$\gamma_s(a) = s \wedge a. \tag{33}$$

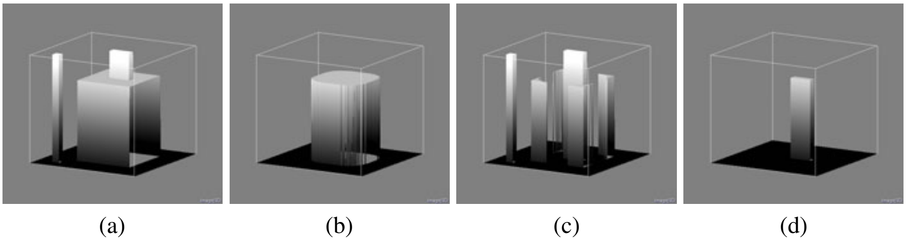


Fig. 2. Hyperconnection derived from structural opening: (a) surface plot of grey-scale image f; (b) structural opening $\gamma(f)$; (c) union of sup-generators $s \in \mathcal{S}$ such that $s \leq f$ but $s \nleq \gamma(f)$; (d) the non-zero infimum of (b) and (c), typical for a hyperconnection, rather than a connection

This is an opening because it is idempotent, anti-extensive and increasing. Now let $t \in \mathcal{S}$ such that $s \leq t$. Obviously $\mathcal{H}^*_{\gamma_s}(t) = \{t\}$ because t is the only hyperconnected component of t. The only hyperconnected component of $\gamma_s(t)$ is s and

$$\mathcal{H}^*_{\gamma_s}(\gamma_s(t)) = \{s\} \nsubseteq \mathcal{H}^*_{\gamma_s}(t), \tag{34}$$

and therefore γ_s is not a hyperconnected filter according to Def. 8.

However, consider the subset of all openings on $\mathcal{L}$ which treats all sup-generators equally, in the sense that either

$$\mathcal{S} \subset \mathrm{Inv}(\gamma) \quad \text{or} \quad \gamma(s) = \mathbf{0} \quad \forall s \in \mathcal{S}. \tag{35}$$

The first case only holds for the identity operator, because if all sup-generators are in $\mathrm{Inv}(\gamma)$, so are all elements of $\mathcal{L}$. The identity operator is obviously hyperconnected. The second case holds in all openings that only preserve substructures in an object larger than any single sup-generator (remember $\mathcal{S} \in \mathcal{P}_{csc}(\mathcal{L})$). In particular, in the lattice of grey-scale images it holds for all translation and grey-scale invariant openings not equal to the identity operator. We will call such openings $\mathcal{S}$*-rejecting* openings.

Theorem 4 ($\mathcal{S}$-rejecting openings). *All $\mathcal{S}$-rejecting openings γ on any complete lattice $\mathcal{L}$, sup-generated by $\mathcal{S} \in \mathcal{P}_{csc}(\mathcal{L})$ are hyperconnected.*

Proof. If $x \in \mathcal{S}$, we have $\gamma(x) = \mathbf{0}$, because γ is $\mathcal{S}$-rejecting, and $\mathcal{H}^*_\gamma(\gamma(x)) = \emptyset$.

For any $x \in \mathcal{L} \setminus \mathcal{S}$, we have either $\gamma(x) = \mathbf{0}$, or $\gamma(x) \in \mathrm{Inv}(\gamma) \setminus \{\mathbf{0}\}$. In the first case we have the same situation as above, and in the second case $h_0 = \gamma(x)$ is the sole hyperconnected component of $\gamma(x)$. Because $h_0 \in \mathcal{H}^*_\gamma(x)$, γ is hyperconnected.

The above hyperconnected openings can be described as attribute filters in which the criterion $\Lambda : \mathcal{H}_\gamma \to \{\mathit{false}, \mathit{true}\}$ is

$$\Lambda(h) = (h \in \mathrm{Inv}(\gamma)). \tag{36}$$

4 Conclusion

In this paper I have shown how to extend the theory of hyperconnections to complete lattices. It was shown that we must require hyperconnections to be chain complete in the sense of [3], in the infinite case, in order to guarantee the existence of hyperconnected components.

Furthermore, a large family of openings, well beyond the cases discussed in [10,5,11] can be shown to be hyperconnected attribute filters. However, though we can write the operator in this way, it is not particularly useful, because the attribute criterion is defined in terms of $\mathrm{Inv}(\gamma)$, which is most easily checked by computing $\gamma(h)$, which is rather circular. Also, the generation of a cover consisting of a single huge entity plus a few details filled in by sup-generators might also be thought of as unsatisfactory.

However, given that there exists a hyperconnection describing any given $\mathcal{S}$-rejecting opening, it should in principle be possible to find a smallest hyperconnection $\mathcal{H}^0_\gamma \leq \mathcal{H}_\gamma$ which describes the opening under study most efficiently. The smallest possible one would be $\{\mathbf{0}\} \cup \mathcal{S}$, which is impractical, because the acceptance or rejection of the hyperconnected components by the filter would not just rely on the components themselves, but on their context as well. An attribute filter using this description would not be adjacency stable [2]. Somewhere between $\{\mathbf{0}\} \cup \mathcal{S}$ and $\mathcal{H}_\gamma$ lies an optimum hyperconnection hyperconnection $\mathcal{H}^0_\gamma$ which is the smallest one which allows description of γ as an adjacency stable, hyperconnected, attribute filter. This hyperconnection would be the *characteristic hyperconnection* of γ. For structural openings, path openings, and viscous openings, these characteristic hyperconnections have been found in [10, 11]. For others, we must still obtain them.

References

1. Braga-Neto, U., Goutsias, J.: A multiscale approach to connectivity. Comp. Vis. Image Understand 89, 70–107 (2003)
2. Crespo, J.: Adjacency stable connected operators and set levelings. Image Vis. Comput. 28(10), 1483–1490 (2010)
3. Markowsky, G.: Chain-complete posets and directed sets with applications. Algebra Universalis 6, 53–68 (1976), `http://dx.doi.org/10.1007/BF02485815`
4. Nempont, O., Atif, J., Angelini, E., Bloch, I.: A new fuzzy connectivity measure for fuzzy sets. J. Math. Imag. Vis. 34, 107–136 (2009)
5. Ouzounis, G.K., Wilkinson, M.H.F.: Hyperconnected attribute filters based on k-flat zones. IEEE Trans. Pattern Anal. Mach. Intell. 33(2), 224–239 (2011)
6. Ronse, C., Serra, J.: Geodesy and connectivity in lattices. Fundamenta Informaticae 46(4), 349–395 (2001)
7. Serra, J.: Introduction to morphological filters. In: Serra, J. (ed.) Image Analysis and Mathematical Morphology. Theoretical Advances, vol. II, ch. 5, pp. 101–114. Academic Press, London (1988)
8. Serra, J.: Connectivity on complete lattices. J. Math. Imag. Vis. 9(3), 231–251 (1998)
9. Serra, J.: Viscous lattices. J. Math. Imag. Vis. 22(2-3), 269–282 (2005)
10. Wilkinson, M.H.F.: An axiomatic approach to hyperconnectivity. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 35–46. Springer, Heidelberg (2009)
11. Wilkinson, M.H.F.: Hyperconnectivity, attribute-space connectivity and path openings: Theoretical relationships. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 47–58. Springer, Heidelberg (2009)

Toward a New Axiomatic for Hyper-Connections

Benjamin Perret[1], Sébastien Lefèvre[2], and Christophe Collet[1]

[1] Image Science, Computer Science and Remote Sensing Laboratory (LSIIT, UMR 7005, University of Strasbourg-CNRS). Address: LSIIT, Pôle API, Bd Sébastien Brant, BP 10413, 67412 Illkirch Cedex France
{bperret,c.collet}@unistra.fr

[2] VALORIA Laboratory, University of South Brittany. Address: VALORIA Lab, Campus de Tohannic, BP 573, 56017 Vannes Cedex, France
sebastien.lefevre@univ-ubs.fr

Abstract. We propose a new class of hyper-connections in order to improve the consistency of hyper-connected filters and to simplify their design. Our idea relies on the principle that the decomposition of an image into h-components must be necessary and sufficient. We propose a set of three equivalent axioms to achieve this goal. We show that an existing h-connection already fulfils these properties and we propose a new h-connection based on flat functions that also fulfils these axioms. Finally we show that this new class brings several new interesting properties that simplify the use of h-connections and guarantee the consistency of h-connected filters as they ensure that: 1) every deletion of image components will effectively modify the filtered image 2) a deleted component can not re-appear in the filtered image.

1 Introduction

Connections in image processing describe how pixels can be grouped together according to their spatial relationships and/or their gray level values. In recent years, several works were devoted to the development of new theories of connections among which hyper-connection (h-connection) [1,2,3] is a very promising notion. Contrary to traditional connections (like set connections), h-connections allow to decompose images into intersecting components.

In this paper, we investigate the problem of the consistency of connected filters (edge-preserving filters) based on h-connections. The usual way to produce a h-connected filter on an image is based on the following three steps: 1) consider the set of h-connected components of this image, 2) select a subset of theses components and 3) reconstruct an image using the select h-components. In this scheme, one expect that the result image contains the selected h-components and only those ones. But in the current theory, we can easily show that the result will contain the selected h-connected components but may also contain other components. These unwanted components are in fact h-components of the original image that have not been selected during the 2nd step (see proposition 2 further). This effect can be circumvented by the addition of a new property that relies on the idea that the decomposition of an image into h-components must

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 85–95, 2011.

be necessary and sufficient to describe the image. We think that this property is fundamental for some image processing tasks as it implies that the decomposition into h-components completely describes the whole image (sufficient) and that none of its components is useless (necessary). These requirements enforce the consistency of the h-connected filters as they ensure that: 1) every deletion of image components will effectively modify the filtered image 2) a deleted component can not re-appear in the filtered image. From a more formal point of view, the sufficiency and necessity conditions can be formulated like this: the supremum of the h-components of an image must be equal to the whole image (sufficient) and the supremum of a family strictly included in the family of h-components must be strictly included in the image (necessary).

In this article we propose a set of three axioms to achieve this goal and we show that they are indeed equivalent. We show that an existing h-connection (the functions with a unique maximum) already fulfills these axioms and we propose a new h-connection based on flat functions, that may be seen as a generalization of the flat zones, that also fulfills these axioms. Finally we show that these new axioms bring several new interesting properties that simplify the use of h-connections and guarantee the consistency of h-connected filters.

This article is organized as follows: section 2 gives some preliminary definitions, section 3 presents the theory of h-connections and establishes some new results, section 4 presents our new theoretical developments and finally section 5 concludes this work.

2 Lattices and Connections

This section gives some preliminary definitions about lattices and set connections. In the sequel, sets and families are written in capital letters while elements of a set are written in lower-case letters. The logical conjunction (respectively disjunction) is noted $\wedge$ (respectively $\vee$).

A lattice $(\mathcal{L}, \leq, \bigvee, \bigwedge)$ is composed of a non-empty set $\mathcal{L}$ with a partial order $\leq$ and two operators: a supremum $\bigvee$ and an infimum $\bigwedge$. In the following, we consider only complete lattices where every non empty family $A \subseteq \mathcal{L}$ has a supremum $\bigvee A$ and an infimum $\bigwedge A$ in $\mathcal{L}$ and we note $\top$ the greatest element ($\bigvee \mathcal{L} = \top$) and $\bot$ the least element ($\bigwedge \mathcal{L} = \bot$) of the lattice. A set $\mathcal{S} \subseteq \mathcal{L}$ such that $\bot \notin \mathcal{S}$ is called a *sup-generating family* of $\mathcal{L}$ if every element of $\mathcal{L}$ can be written as the supremum of elements of $\mathcal{S}$ ($\forall a \in \mathcal{L},\ \exists B \subseteq \mathcal{S},\ a = \bigvee B$). The elements of $\mathcal{S}$ are called *sup-generators*. Important examples of lattices are:

- the extended real line $\overline{\mathbb{R}} = \mathbb{R} \bigcup \{-\infty, +\infty\}$, which is a complete chain under the usual order, infimum, and supremum. It is sup-generated by $\mathbb{R}$.
- the set of all subsets of a set E (noted $\mathcal{P}(E)$), which is a complete lattice with the partial order defined by the inclusion relation, and the infimum and supremum given by the set intersection and union. It is sup-generated by the singletons of E.

- the set of functions from a set E into a lattice $\mathcal{L}$ (noted $\mathcal{L}^E$), which is a lattice under the pointwise order $\forall f, g \in \mathcal{L}^E, f \leq g \Leftrightarrow \forall x \in E, f(x) \leq_{\mathcal{L}} g(x)$. The infimum and supremum are similarly defined by a pointwise application of the infimum $\bigwedge_{\mathcal{L}}$ and supremum $\bigvee_{\mathcal{L}}$ of the underlying lattice. $\mathcal{L}^E$ is sup-generated by the pulses: the functions δ_x^t, $\forall x \in E$, $\forall t \in \mathcal{L}$, $t \neq \bot$ defined by $\forall y \in E$, $\delta_x^t(y) = t$ if $x = y$, $\bot$ otherwise. The properties of $\mathcal{L}^E$ depend of the properties of the underlying lattice $\mathcal{L}$.

A connection on a lattice $\mathcal{L}$, furnished with a sup-generating family $\mathcal{S}$, is a family $\mathcal{C} \subseteq \mathcal{L}$ composed of the connected elements of $\mathcal{L}$. Formally we say that $\mathcal{C}$ is a connection if [4]:

1. $\bot \in \mathcal{C}$: the least element is connected;
2. $\forall s \in \mathcal{S}$, $s \in \mathcal{C}$: the sup-generators are connected;
3. $\forall A \subseteq \mathcal{C}$, $\bigwedge A \neq \bot \Rightarrow \bigvee A \in \mathcal{C}$, the supremum of intersecting connected elements is connected.

3 Hyper-Connection

We now present the theory of h-connections for the finite case, we give the definition of a new h-connection and we establish two new properties. The extension of the theory to infinite (countable or not) lattices is still an open problem [3] and is not discussed here. Being given a lattice $\mathcal{L}$ with a sup-generating family $\mathcal{S}$, an h-connection $\mathcal{C}^+$ on $\mathcal{L}$ is a subset of $\mathcal{L}$ verifying the following conditions [1]:

1. $\bot \in \mathcal{C}^+$: the least element is h-connected;
2. $\forall s \in \mathcal{S}$, $s \in \mathcal{C}^+$: the points (i.e. the sup-generators) are h-connected;
3. $\forall A \subseteq \mathcal{C}^+$, $\bowtie A \Rightarrow \bigvee A \in \mathcal{C}^+$, the supremum of overlapping h-connected elements is h-connected.

where $\bowtie$ is a predicate on $\mathcal{P}(\mathcal{L})$ called the overlap criterion. This predicate must be decreasing: $\forall A \in \mathcal{P}(\mathcal{L})$, $\forall B \in \mathcal{P}(\mathcal{L})$, $A \subseteq B$, $\not\bowtie A \Rightarrow \not\bowtie B$ meaning that a family of non overlapping elements cannot become overlapping by the addition of new elements. Note that a connection is a particular case of h-connections with the overlap criterion defined by: $\forall A \in \mathcal{P}(\mathcal{L})\,,\ \bowtie A \Leftrightarrow \bigwedge A \neq \bot$.

A simple example of h-connection is made of functions with a unique maximum [1,2]. This h-connection is defined on the lattice of functions $\mathcal{L}^E$ and it is based on a primary set connection $\mathcal{C}_o$ on E. Then, one says that a function has a unique maximum if it is connected at all levels with respect to the primary set connections. More formally, we define the set of connected functions by: $\mathcal{C}_m^+ = \left\{ f \in \mathcal{L}^E \,|\, \forall t \in \mathcal{L},\ \overline{f}^t \in \mathcal{C}_o \right\}$, with $\overline{f}^t = \{p \in E \,|\, t \leq f(p)\}$ the thresholding of f at level t. The overlap criterion can be defined as: $\forall \{f_i\} \subseteq \mathcal{L}^E, \bowtie_m \{f_i\} \Leftrightarrow \forall t \in \mathcal{L}$, $U_t = \emptyset$ or $\bigcap U_t \neq \emptyset$ with $U_t = \left\{ \overline{f_i}^t \,|\, \overline{f_i}^t \neq \emptyset \right\}$.

Another new example of h-connection is given by the set of all flat functions. Let $\mathcal{L}^E$ be the image space, let $\mathcal{C}_o$ be a primary set connection on $\mathcal{P}(E)$, the set of flat functions is defined by:

$$\mathcal{C}_p^+ = \left\{ f_{C,t} \in \mathcal{L}^E \,|\, C \in \mathcal{C}_o, t \in \mathcal{L} \right\} \tag{1}$$

with $\forall x \in E,\ f_{C,t}(x) = t$ if $x \in C$, $\bot$ otherwise. Then the overlap criterion $\bowtie_p$ is defined by:

$$\forall \{g_{i\in I}\} \subseteq \mathcal{C}_p^+, \bowtie_p (\{g_i\}) = \begin{cases} \text{true} & \text{if } U \neq \emptyset \text{ and } \forall i,j \in I, \forall x \in U, g_i(x) = g_j(x) \\ \text{false} & \text{otherwise} \end{cases} \tag{2}$$

with $U = \bigcap_{i\in I} \operatorname{supp}(g_i)$ and $\operatorname{supp}(g_i) = \{p \in E \,|\, g_i(p) \neq \bot\}$ is the support of the function g_i. Flat functions are overlapping if their supports intersect and if they have the same value on this intersection. Figure 1 shows examples of overlapping and non-overlapping flat functions according to $\bowtie_p$. Contrary to the usual definition of flat zones (the largest connected component such that the function is constant), the h-connected flat zones can spread under other flat zones that have a higher level. Several flat zones according to the usual definition can then be represented by a unique h-connected flat function.

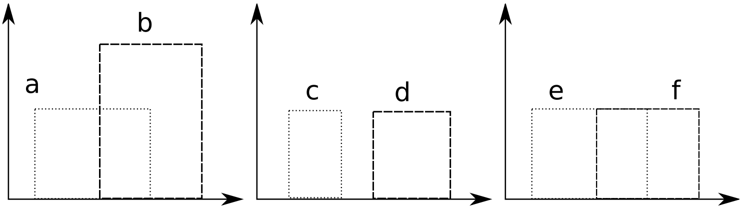


Fig. 1. The overlap criterion $\bowtie_p$. Functions a and b are not overlapping because they don't have the same value on their intersection. Functions c and d are not overlapping because their supports do not intersect. Functions e and f overlap.

Proposition 1. *$(\mathcal{C}_p^+, \bowtie_p)$ is an h-connection of $\mathcal{L}^E$ (see Figure 2).*

Proof. We must show that the three axioms of the h-connection are verified by $(\mathcal{C}_p^+, \bowtie_p)$. First note that $\bowtie_p$ is trivially decreasing.

1) By definition of set connections, $\emptyset \in \mathcal{C}_o$, $\forall t \in \mathcal{L}$, $f_{\emptyset,t} = \bot \in \mathcal{C}_p^+$.

2) By definition of set connections, the singletons of E belong to $\mathcal{C}_o$, thus, $\forall x \in E$, $\forall t \in \mathcal{L}$, the $f_{\{x\},t} = \delta_{x,t} \in \mathcal{C}_p^+$ are the pulses of $\mathcal{L}^E$ which are a sup-generating family $\mathcal{S}$ of $\mathcal{L}^E$.

3) Let $\{a_{i\in I}\}$ be a family of $\mathcal{C}_p^+$, let also $U = \bigcap_{i\in I} \operatorname{supp}(a_i)$, and assume that $U \neq \emptyset$ and $\forall i,j \in I$, $\forall x \in U$, $a_i(x) = a_j(x)$ then we must show that $\bigvee a_i \in \mathcal{C}_p^+$. From property 3) of set connections, if $U \neq \emptyset$ then $\bigcup_{i\in I} \operatorname{supp}(a_i) = R \in \mathcal{C}_o$, then from the definition of $\mathcal{C}_p^+$ and from the condition $\forall i,j \in I$, $\forall x \in U$, $a_i(x) = a_j(x)$, all a_i have the same value t on their supports, thus, $\bigvee a_i = f_{R,t} \in \mathcal{C}_p^+$.

Hyper-connected operators and openings. H-operators [3] are applications from $\mathcal{L}$ into $\mathcal{P}(\mathcal{L})$ which extract the h-components marked by a sup-generator. Formally, being given a sup-generator $s \in \mathcal{S}$, we define the h-operator by:

$$\forall a \in \mathcal{L},\ \gamma_s^*(a) = \left\{h \in \mathcal{C}^+ \,|\, s \leq h \leq a, \forall g \in \mathcal{C}^+, h \leq g \leq a \Rightarrow g = h\right\} \tag{3}$$

The h-components are maximal h-connected elements and the h-components of $a \in \mathcal{L}$ are given by:

$$\gamma^* \left(a\right) = \bigcup_{s \in \mathcal{S}} \gamma_s^* \left(a\right) \tag{4}$$

The h-opening marked by s is the supremum of the h-components of a above s:

$$\forall a \in \mathcal{L}, \quad \gamma_s \left(a\right) = \bigvee \gamma_s^* \left(a\right) \tag{5}$$

The two following properties hold [2]:

1. $a = \bigvee \gamma^* \left(a\right)$: an element is the supremum of its h-components;
2. $\forall b, c \in \gamma^* \left(a\right)$, $b \neq c \Rightarrow b \not\bowtie c$: two h-components do not overlap.

Z-operators. In [2], the authors propose the interesting notion of *z-operators*. Let $a \in \mathcal{L}$, the equivalence relation $\overset{a}{\sim}$ on $\mathcal{S}$ is defined by:

$$\forall b, c \in \mathcal{S}, \quad b \overset{a}{\sim} c \Leftrightarrow \gamma_b \left(a\right) = \gamma_c \left(a\right) \tag{6}$$

Then, the z-operator is defined as the supremum of an equivalence class of $\overset{a}{\sim}$:

$$\forall s \in \mathcal{S}, \quad \zeta_s \left(a\right) = \bigvee \left\{ b \in \mathcal{S} \,|\, b \overset{a}{\sim} s \right\} \tag{7}$$

Finally, the set of all z-zones of a is noted:

$$\zeta \left(a\right) = \left\{ \zeta_s \left(a\right) \,|\, \forall s \in \mathcal{S} \right\} \tag{8}$$

The authors of [2] have chosen to base the z-operators on the h-connected openings, but we can also define them in terms of h-connected operators leading to a slightly different definition. Let $\overset{a*}{\sim}$ be the equivalence relation defined by:

$$\forall b, c \in \mathcal{S}, \quad b \overset{a*}{\sim} c \Leftrightarrow \gamma_b^* \left(a\right) = \gamma_c^* \left(a\right) \tag{9}$$

The alternative z-operator* is the supremum of an equivalence class of $\overset{a*}{\sim}$:

$$\forall s \in \mathcal{S}, \quad \zeta_s^* \left(a\right) = \bigvee \left\{ b \in \mathcal{S} \,|\, b \overset{a*}{\sim} s \right\} \tag{10}$$

Generally, as h-components can be intersecting, $\zeta_s \left(a\right) \neq \zeta_s^* \left(a\right)$. Consider the set $E = \{a, b, c\}$ and the h-connection $\{\emptyset, \{a\}, \{b\}, \{c\}, \{ab\}, \{bc\}, \{ac\}\}$ on $\mathcal{P}\left(E\right)$ with the overlap criterion defined as false (two elements are never overlapping, this is a valid criterion since it is trivially decreasing). Then, the h-components of E are $\{\{ab\}, \{bc\}, \{ac\}\}$. Thus, we have $\gamma_a \left(E\right) = \gamma_b \left(E\right) = \gamma_c \left(E\right) = E$ and $\zeta_a \left(E\right) = \zeta_b \left(E\right) = \zeta_c \left(E\right) = E$. But $\gamma_a^* \left(E\right) = \{\{ab\}, \{ac\}\}$, $\gamma_b^* \left(E\right) = \{\{ab\}, \{bc\}\}$ and $\gamma_c^* \left(E\right) = \{\{ac\}, \{bc\}\}$ thus $\zeta_a^* \left(E\right) = \{a\} \neq \zeta_a \left(E\right)$. In section 4, we show that under certain conditions, we have $\forall a \in \mathcal{L}, \forall s \in \mathcal{S}, \zeta_s \left(a\right) = \zeta_s^* \left(a\right)$. Figure 2 shows an example of function decomposition with the z-operators and the h-connection of functions having a unique maximum.

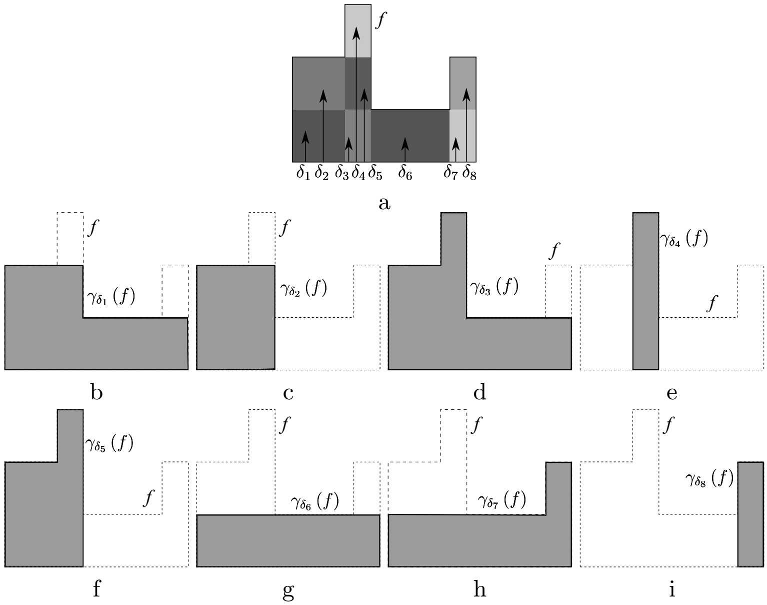


Fig. 2. Example of decomposition of the function f using the h-connection of flat functions. Figure (a) shows the function f, 8 pulses $\delta_1, \ldots, \delta_8$ that are representative of the 8 equivalence classes of the sup-generating family according to $\overset{f}{\sim}$. In image (a), each zone of f marked by the pulses δ_i, corresponds to an equivalence class and is thus the result of a z-operator $\zeta_{\delta_i}(f)$. Figures (b) to (i) show the results of the h-openings of f marked by the δ_i. One can note that figures (c), (e), (g) and (i) are the h-components of f while the results of the h-opening marked by δ_1, δ_3, δ_5 and δ_7 represented in figures (b), (d), (f) and (h) are not h-connected.

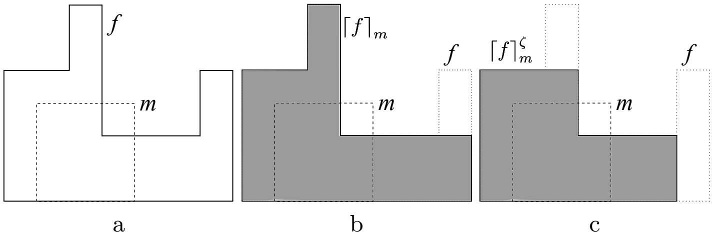


Fig. 3. Example of h-reconstruction with the h-connection of flat functions. Figure (a) shows the function f and the marker m. The figure (b) represents the result of the h-reconstruction $\lceil f \rceil_m$ of f by m proposed in [2]. Figure (c) represents the result $\lceil f \rceil_m^\zeta$ of f by m with our definition.

H-reconstruction. Being given a marker $m \in \mathcal{L}$, the hyper-reconstruction (h-reconstruction) of $a \in \mathcal{L}$ marked by m is defined by [2]:

$$\lceil a \rceil_m = \bigvee_{s \in \mathcal{S}, s \leq m} \gamma_s (a) \tag{11}$$

We propose another definition based on the alternative z-operators:

$$\lceil a \rceil_m^\zeta = \bigvee_{s \in \mathcal{S}, s \leq m} \zeta_s (a) \tag{12}$$

The latter is more flexible, the two approaches are compared in Figure 3.

New properties. We establish here two properties that will help us in following proofs. Let $a \in \mathcal{L}$:

Proposition 2. *$\forall \{h_i\} \subseteq \gamma^* (a)$, $\{h_i\} \subseteq \gamma^* (\bigvee \{h_i\})$*

Proof. Let a in $\mathcal{L}$, $\{h_i\} \subseteq \gamma^* (a)$ and $b = \bigvee \{h_i\}$, for all i , we have $h_i \leq b$ and $b \leq a$. Suppose that there exists some $h \in \mathcal{C}^+$ such that $h_i \leq h \leq b \leq a$. But by definition of $\gamma^* (a)$ and as h_i belongs to $\gamma^* (a)$, we have $h_i = h$ showing that $h_i \in \gamma^* (b)$. Thus we have $\{h_i\} \subseteq \gamma^* (\bigvee \{h_i\})$.

This property that was already known under a different form in [2] states that: being given a family of h-components, the h-components of its supremum is a superset of the given family. Those two families are generally not equal, take for example the h-connection given in section 3: the h-components of the element $\{abc\}$ are $\{\{ab\}, \{bc\}, \{ac\}\}$. Now, consider the subfamily $\{\{ab\}, \{bc\}\}$, the supremum of this family is equal to $\{abc\}$ and the h-components of this supremum is a strict superset of the family.

Proposition 3. *$\forall s \in \mathcal{S}$, $\forall h' \in \gamma^* (a)$, if $s \leq h'$ and $\gamma_s (a) = h \in \mathcal{C}^+$ then $h = h'$.*

Proof. Let s in $\mathcal{S}$, $h \in \mathcal{C}^+$ such that $\gamma_s (a) = h$ and $h' \in \gamma^* (a)$ such that $s \leq h'$. We have $h' \leq \bigvee \{g \in \gamma^* (a) \mid s \leq g\} = h$, so $s \leq h' \leq h \leq a$ and by definition of h-components $h = h'$.

If the h-opening of a marked by the sup-generator s is an h-component then there is no other h-component of a above s.

In the following, we assume the existence of a complete lattice $\mathcal{L}$ with the sup-generating family $\mathcal{S}$ and the h-connection $\mathcal{C}^+$ such that the number of h-components of each element of $\mathcal{L}$ is finite.

4 Toward a New Class of Hyper-Connections

Despite the successful developments of h-connections in recent works [3,5,6,7,8,2], the theory of h-connections is still not satisfactory. All of the successes obtained have been based on specific h-connections, whereas only a few general properties have been established. This lack of theoretical results comes from the very broad definition of h-connections. The third axiom (the overlap criterion) is especially problematic. On one hand, this axiom is satisfactory because it formalizes the intuitive union based approach. On the other hand, the definition of the overlap criterion is so wide that, in practice, this third axiom does not bring any meaningful property to the h-connection (for example, consider the overlap criterion which is always false, and the third axiom is never applicable). In fact one can note that, for any set $\mathcal{L}$ verifying axioms 1 and 2 of h-connections, there exists a set of possible overlap criteria $\mathcal{O}$ such that $\forall \bowtie \in \mathcal{O}$, the couple $(\mathcal{L}, \bowtie)$ is an h-connection. The set $\mathcal{O}$ is partially ordered (given two overlap criteria $\bowtie_1$ and $\bowtie_2$, we say that $\bowtie_1$ is lower than $\bowtie_2$ if $\forall A \in \mathcal{P}(\mathcal{L})$, $\bowtie_1 A \Rightarrow \bowtie_2 A$) and it owns a least and a greatest element. The least overlap criterion (noted $\bowtie_\perp$) is the "always false criterion" while the greatest overlap criterion (noted $\bowtie_\top$) is given by $\forall A \subseteq \mathcal{C}^+$, $\bowtie A \Leftrightarrow (\forall B \subseteq A,\ \bigvee B \in \mathcal{C}^+)$. Usually, one tends to choose an overlap criterion close to the greatest one (the most constraining one) but rarely reaches it [1,2,7,5,6]. One can also observe that the greatest overlap criterion $\bowtie_\top$ is redundant with the definition of the set of h-connected elements $\mathcal{C}^+$: on one hand $\bowtie_\top$ is completely defined by the definition of $\mathcal{C}^+$ and on the other hand, given $\bowtie_\top$, $\mathcal{C}^+$ can be recovered by computing the transitive closure conditionally to $\bowtie_\top$ of the set of sup-generators (the limit of the sequence $(\mathcal{C}_n^+)$ defined recursively by $\mathcal{C}_0^+ = \mathcal{S}$ and $\forall n \in \mathbb{N}^*$, $\mathcal{C}_n^+ = \mathcal{C}_{n-1}^+ \cup \left\{x \in \mathcal{L} \,|\, \exists A \in \mathcal{P}\left(\mathcal{C}_{n-1}^+\right),\ \bowtie A,\ x = \bigvee A\right\}$).

To harden the definition of h-connections, we propose the fundamental property that an h-connection must provide for each element a decomposition in h-components that is necessary and sufficient. With the current theory it is clear that the decomposition in h-components is sufficient in the sense that an element is equal to the supremum of its h-components. But in general this decomposition is not necessary in the sense that only a subset of the h-components of an element can be necessary to retrieve the element by supremum.

More formally, one can define the following properties $\forall a \in \mathcal{L}$:

P-1 – $\forall h' \in \gamma^*(a),\ \bigvee \{h \in \gamma^*(a) \,|\, h \neq h'\} < \bigvee \gamma^*(a)$: all h-components of an element are necessary to describe the element.

P-2 – $\forall \{h_i\} \subseteq \gamma^*(a),\ \forall h \in \gamma^*(a), h \leq \bigvee \{h_i\} \Rightarrow h \in \{h_i\}$: an h-component cannot be covered by other h-components.

P-3 – $\forall h \in \gamma^*(a),\ \exists s \in \mathcal{S},\ \gamma_s^*(a) = h$: each h-component of an element can be individually retrieved through an h-opening.

Then, the following proposition holds:

Proposition 4. *Properties P-1, P-2, and P-3 are equivalent. In the following, we will say that a connection is accessible if the previous properties hold.*

Proof. We first show P-2⇒P-1. Let $\{h_i\} \subseteq \gamma^*(a)$, and assume that $\exists h' \in \{h_i\}$ such that $\bigvee \{h \in \gamma^*(a) \,|\, h \neq h'\} = \bigvee \gamma^*(a)$. Let $F = \bigvee \{h \in \gamma^*(a) \,|\, h \neq h'\}$, one have $h' \leq \bigvee \gamma^*(a) = \bigvee F$ but that contradicts P-2 since $h' \notin F$.

Then, we show P-1⇒P-2: let $\{h_i\} \subset \gamma^*(a)$ and assume that there exists $h \in \gamma^*(a)$ such that $h \notin \{h_i\}$ and $h \leq \bigvee \{h_i\}$. Let $b = \bigvee \{h_i\} = \bigvee (\{h_i\} \cup \{h\})$, we have by proposition 2 that $h \in \gamma^* (b)$ and that $\{h_i\} \subseteq \gamma^* (b)$. And then we have $\bigvee \{h' \in \gamma^* (b) \mid h' \neq h\} = \bigvee \{h_i\} = b$ which contradicts P-1.

Now P-3⇒P-2: let $\{h_i\} \subseteq \gamma^* (a)$ and $h \in \gamma^* (a)$. Assume that $h \leq \bigvee \{h_i\}$ then, by P-3, there exists $s \in \mathcal{S}$ such that $\gamma_s (a) = h$ and $s \leq h \leq \bigvee \{h_i\}$. By assumption, the family $\gamma^* (a)$ is finite, so there exists j such that $s \leq h_j$ and from proposition 3 we can say that $h = h_j$ and thus h belongs to $\{h_i\}$.

And finally, P-1⇒P-3: let $h \in \gamma^* (a)$, P-1 implies that $\bigvee B < a$ with $B = \{h' \in \gamma^* (a) \mid h' \neq h\}$. Now, consider the smallest family $\{s_i\} \subseteq \mathcal{S}$ such that $\bigvee (B \cup \{s_i\}) = a$. Let s be an element of $\{s_i\}$, we have for all $h' \in B$, $s \nleq h'$ and $s \leq h$ thus $\gamma_s (a) = h$.

P-1 and P-2 are two direct formulations of the "necessity" condition, the first one from a global point of view, the second from a local point of view. P-3 shows that this condition naturally comes to the notion of special groups of sup-generators that give access to a unique h-component through an h-opening. From the image processing point of view, this means that every h-component of an image can be selected individually with a simple opening.

Proposition 5. *The h-connection* $(\mathcal{C}_m^+, \bowtie_m)$ *of functions with a unique maximum is accessible.*

Proof. We show that $\mathcal{C}_m^+$ verifies the property P-3. Let $a \in \mathcal{L}^E$, $h \in \gamma^* (a)$ and $x \in E$ such that $h(x) = \max_{y \in E} h(y)$ ($h(x)$ is in the maximum of h). Let the pulse $\delta_{x,h(x)}$ and we show that $\gamma^*_{\delta_{x,h(x)}} (a) = \{h\}$.

The inclusion $\{h\} \subseteq \gamma^*_{\delta_{x,h(x)}} (a)$ is direct as $\delta_{x,h(x)} \leq h$ and $h \in \gamma^* (a)$.

We now show the inverse inclusion $\gamma^*_{\delta_{x,h(x)}} (a) \subseteq \{h\}$. Let $h' \in \gamma^*_{\delta_{x,h(x)}} (a)$. We start by showing that $\bowtie_m (\{h, h'\})$ is true. Let $t \in \mathcal{L}$, we have: if $t \leq h(x)$ then $\overline{h}^{\urcorner t} \neq \emptyset$ and $\overline{h}^{\urcorner t'} \neq \emptyset$ (as $\delta_{x,h(x)} \leq h'$), else if $t > h(x)$ then $\overline{h}^{\urcorner t} = \emptyset$ ($h(x)$ is the maximal height of h). And yet $\forall t \in \mathcal{L}$, $t \leq h(x)$, we have $x \in \overline{h}^{\urcorner t} \cap \overline{h'}^{\urcorner t} \neq \emptyset$. So $\bowtie_m (\{h, h'\})$ is true. In consequence, $\bigvee \{h, h'\} \in \mathcal{C}_m^+$, but $h, h' \leq a$ so $\bigvee \{h, h'\} \leq a$, and as h and h' are h-connected components of a, we have either $h = h'$ or h and h' are not comparable. Assume that h and h' are not comparable, then $h < \bigvee \{h, h'\}$, $h' < \bigvee \{h, h'\}$, and as $\bigvee \{h, h'\} \leq a$, h and h' cannot be h-connected components of a. Thus we have $h = h'$ and $\gamma^*_{\delta_{x,h(x)}} (a) \subseteq \{h\}$.

Thus, we have the double inclusion and $\gamma^*_{\delta_{x,h(x)}} (a) = \{h\}$. Every h-connected component of a can be obtained by an h-opening. The h-connection $\mathcal{C}_m^+$ verifies property P-3 and is accessible.

Proposition 6. *The h-connection* $(\mathcal{C}_p^+, \bowtie_p)$ *of flat functions is accessible.*

Proof. The demonstration is similar to the previous one using the opening $\gamma^*_{\delta_{x,h(x)}} (a) = \{h\}$ with $x \in E$ such that $h(x) = a(y)$.

Accessible h-connections also have stronger properties:

Proposition 7. *If* $\mathcal{C}^+$ *is accessible, being given a family of h-components, the h-components of its supremum is the same family:* $\forall \{h_i\} \subseteq \gamma^* (a)$, $\{h_i\} = \gamma^* (\bigvee \{h_i\})$

Proof. We already have the first inclusion $\{h_i\} \subseteq \gamma^* \left(\bigvee \{h_i\}\right)$ by proposition 2. The second inclusion $\gamma^* \left(\bigvee \{h_i\}\right) \subseteq \{h_i\}$ is directly given by P-2: let $h \in \gamma^* \left(\bigvee \{h_i\}\right)$, as we have $h \leq \bigvee \{h_i\}$, P-2 says that h belongs to $\{h_i\}$. Finally, the double inclusion proves that $\{h_i\} = \gamma^* \left(\bigvee \{h_i\}\right)$.

Compared to proposition 2, this version is harder and it ensures that by selecting a family of h-components, its reconstruction by the supremum operator will not introduce new h-components. From the image processing point of view, this ensures that, when performing an h-connected filtering, a deleted h-component cannot re-appear after the reconstruction of the selected h-components. Another interesting property of accessible h-connections concerns z-operators:

Proposition 8. *If $\mathcal{C}^+$ is accessible, the equivalence relations $\overset{a}{\sim}$ and $\overset{a*}{\sim}$ are equivalent: $\forall s_1, s_2 \in \mathcal{S}$, $\left(s_1 \overset{a}{\sim} s_2\right) \Leftrightarrow \left(s_1 \overset{a*}{\sim} s_2\right)$.*

Proof. First, we show that $\forall s_1, s_2 \in \mathcal{S}$, $s_1 \overset{a*}{\sim} s_2 \Rightarrow s_1 \overset{a}{\sim} s_2$, this part does not need the accessibility property. We have:

$$s_1 \overset{a*}{\sim} s_2 \Leftrightarrow \gamma^*_{s_1}(a) = \gamma^*_{s_2}(a) \Rightarrow \bigvee \gamma^*_{s_1}(a) = \bigvee \gamma^*_{s_2}(a) \Leftrightarrow \gamma_{s_1}(a) = \gamma_{s_2}(a)$$
$$\Leftrightarrow s_1 \overset{a}{\sim} s_2$$

The reverse implication is given by: assume that we have $s_1 \overset{a}{\sim} s_2$, which is equivalent to $\gamma_{s_1}(a) = \gamma_{s_2}(a)$. Let $h \in \gamma^*_{s_1}(a)$, we have $h \leq \gamma_{s_1}(a) = \bigvee \gamma^*_{s_1}(a) = \bigvee \gamma^*_{s_2}(a)$. Then, P-2 implies that $h \in \gamma^*_{s_2}(a)$ and thus $\gamma^*_{s_1}(a) \subseteq \gamma^*_{s_2}(a)$. The same argument is used to prove that $\gamma^*_{s_2}(a) \subseteq \gamma^*_{s_1}(a)$ showing that $\gamma^*_{s_2}(a) = \gamma^*_{s_1}(a)$ which is equivalent to $s_1 \overset{a*}{\sim} s_2$.

This property removes the necessity of operating a choice between the two distinct approaches when dealing with z-operators. Thus it simplifies the process of creating image filters based on z-zones.

5 Conclusion

We have proposed a new class of h-connections motivated by the desirable property that the decomposition of an image into connected components should be necessary and sufficient to describe this image. Due to this evolution, based on three equivalent axioms, we have set out several new important properties for image processing which ensure the consistency and simplify the use of h-connected filters. This theoretical work offers a strong basis for the development of hierarchical representations based on h-connections [9].

The axiomatic of h-connections is still an open problem because it is only valid in the finite case and because its third axiom is not well established. Accessible connexions are certainly too constrained to be the new axiomatic we are looking for, because it excludes some useful operators like structural filters and path filters. Nevertheless, besides being a strong basis for meaningful h-connected filter definition, it is a new step toward the researched axiomatic.

References

1. Serra, J.: Connectivity on complete lattices. JMIV 9(3), 231–251 (1998)
2. Braga-Neto, U., Goutsias, J.: A theoretical tour of connectivity in image processing and analysis. JMIV 19(1), 5–31 (2003)
3. Wilkinson, M.H.F.: An axiomatic approach to hyperconnectivity. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 35–46. Springer, Heidelberg (2009)
4. Serra, J.: Image Analysis and Mathematical Morphology. Theoretical Advances, vol. II. Academic Press, London (1988)
5. Ouzounis, G.K.: Generalized Connected Morphological Operators for Robust Shape Extraction. PhD thesis, University of Groningen (2009)
6. Ouzounis, G.K., Wilkinson, M.: Hyperconnected attribute filters based on k-flat zones. IEEE TPAMI 33(2), 224–239 (2011), doi:10.1109/TPAMI.2010.74
7. Nempont, O., Atif, J., Angelini, E., Bloch, I.: A new fuzzy connectivity measure for fuzzy sets. JMIV 34(2), 107–136 (2009)
8. Wilkinson, M.H.F.: Hyperconnectivity, attribute-space connectivity and path openings: Theoretical relationships. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 47–58. Springer, Heidelberg (2009)
9. Perret, B., Lefèvre, S., Collet, C.: Hyperconnections and hierarchical representations for grayscale and multiband image processing. IEEE TIP (submitted, 2011)

Preventing Chaining through Transitions While Favouring It within Homogeneous Regions

Pierre Soille

IPSC Global Security and Crisis Management Unit
Joint Research Centre, European Commission
T.P. 267, via E. Fermi 2749, I-21027 Ispra, Italy

Abstract. By its very nature, alpha-connectivity is subject to the chaining property of single linkage clustering. Thanks to the selection of appropriate connectivity constraints, connected components that are affected by the chaining through transitions can be invalidated. However, it may happen that (i) a stream of small connected components at the transition between larger components are created and (ii) none of the connected components is matching a desired object whatever the threshold levels associated with the constraints. These two problems are caused by the presence of transitions. In this paper, we characterise transitions in view of their impact on constrained connected paths. We then show that both problems can be addressed simultaneously by either pre-filtering or by introducing a dissimilarity measurement preventing connections through transitions while keeping a definition based on absolute difference measurements.

Keywords: single-linkage clustering, alpha-connectivity, dissimilarity, constrained connectivity, increment operators, smooth connection.

1 Introduction

A connectivity relation between elements of an arbitrary set is an equivalence relation producing a unique partition of this set. The partition cells are equivalence classes that correspond to connected regions of maximal extent called connected components. For example, in image processing, a connectivity relation can be obtained by stating that two pixels of a grey tone image are connected if there exists a path of pixels linking these pixels and such that the grey level absolute difference between adjacent pixels of the path (i.e., weights of the edges of the path) does not exceed a given threshold value [1,2]. The resulting connected components are called quasi-flat [3] or lambda-flat [4] zones in mathematical morphology. This type of connectivity can be expressed in terms of a connective criterion as shown in [5]. The terminologies quasi- and lambda-flat zones are somewhat misleading because the 'flatness condition' set by the threshold value lambda acts only on paths so that it does not forbid the creation of connected components containing adjacent pixels with a local difference larger than the value of lambda. For this reason and because it is explicitly referring to the

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 96–107, 2011.

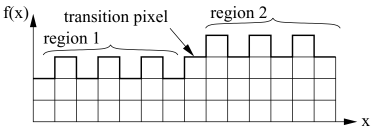


Fig. 1. A 1-dimensional hypothetical signal with a 1 pixel thick transition between two regions characterised by a square signal

notion of connected component, the terminology α-connected component was introduced in [6]. A hierarchy of fine to coarse partitions of the image domain into α-connected components is obtained by increasing the value of the absolute difference threshold parameter α.

It has already been recognised in [2,6] that α-connectivity corresponds to the single linkage clustering which is itself related to the notion of minimum spanning tree [7]. This hierarchical clustering method is known for not being subject to ties (so that it generates unique partitions) but, on the other hand, for being sensitive to the presence of samples of intermediate characteristics. This sensitivity is referred to as the *chaining effect* in [8] but we prefer to call it *chaining through transitions* because the chaining effect itself is related to the very nature of single linkage and is therefore a property rather than an effect. Constrained connectivity addresses this problem by invalidating connected components that are subject to the chaining through transitions. This is achieved by selecting the largest connected components satisfying a series of constraints expressed in terms of logical predicates [9]. However, while this approach succeeds in preventing the selection of α-connected components that were subject to chaining through transitions, it creates a stream of small α-connected components located in image regions called transitions and characterised by a gradual change in brightness from neighbouring dark and bright regions. Moreover, in some situations the formation of connected components that would otherwise be desired is not occurring whatever the considered constraints and their associated threshold values. Take for example the 1-dimensional signal illustrated in Fig. 1 representing two regions characterised by identical square signals up to an offset equal to their amplitude. For this signal, depending on whether the α threshold is less than the amplitude or not, there exists either a α-connected component for each pixel or a unique α-connected component matching all pixels. Therefore, the application of constraints to the α-connected components will never lead to three connected components (one for each region and one for the so-called transition pixel). While the example of Fig. 1 is artificial and extreme, transitions are commonly observed on digital images. Accordingly, image partitioning based on constrained connectivity leads to a string of small connected components at the boundary of larger components or even impede the formation of desirable connected components.

In this paper, after a series of preliminary definitions and notations on hierarchical graph segmentation and constrained connectivity (section 2), a characterisation of transition regions and their impact on α-paths are presented (section

3). Then, methods for handling transitions are detailed (section 4). Before concluding, the experimental section (section 5) shows that the method based on a dissimilarity measure preventing connections through transitions while keeping a definition based on absolute difference measurements is the most effective.

2 Preliminaries

This section presents definitions related to hierarchical graph segmentation in general and then in the context of constrained connectivity.

2.1 Hierarchical Graph Segmentation

A graph[1] is an ordered pair (V, E) comprising a set V of vertices followed by a set E of edges, which are 2-element subsets of V. The operators that return the vertex and edge sets of a graph are denoted by $\mathcal{V}$ and $\mathcal{E}$ respectively. Hence, if $X = (V, E)$ then $V = \mathcal{V}(X)$ and $E = \mathcal{E}(X)$. A graph Y is a subgraph of a graph X if and only if $\mathcal{V}(Y) \subseteq \mathcal{V}(X)$ and $\mathcal{E}(Y) \subseteq \mathcal{E}(X)$. Two subgraphs X_1 and X_2 of a graph X are said to be adjacent if there exists at least one edge $e = \{v_i, v_j\}$ of $\mathcal{E}(X)$ such that $v_i \in \mathcal{V}(X_1)$ and $v_j \in \mathcal{V}(X_2)$. A subgraph Y of a graph X is said to be induced if, for any pair of vertices $\{v_i, v_j\}$ of Y, $\{v_i, v_j\}$ is an edge of Y if and only if $\{v_i, v_j\}$ is an edge of X. If the vertex set of Y is a subset S of $\mathcal{V}(X)$, then Y can be written as $X[S]$ and it is said to be induced by S.

By analogy with the usual definition of a segmentation in image processing [11], we define the segmentation of a connected graph X as a collection of connected induced subgraphs of X realising a partition of the vertices of X and such that there exists a logical predicate P returning true on each subgraph but false on any graph induced by the union of the vertices of adjacent subgraphs. That is, a series of subgraphs X_i of a graph X forms a segmentation of this graph if and only if the following six conditions are met (i) $X_i = X[\mathcal{V}(X_i)]$ for all i, (ii) $\cup_i \mathcal{V}(X_i) = \mathcal{V}(X)$, (iii) $\mathcal{V}(X_i) \cap \mathcal{V}(X_j) = \emptyset$ for all $i \neq j$, (iv) X_i is connected for all i, (v) $P(X_i) =$ true for all i, and (vi) $P(X[\mathcal{V}(X_i) \cup \mathcal{V}(X_j)]) =$ false if X_i and X_j are adjacent.

The coarsest possible segmentation of a graph is the graph itself and its finest possible segmentation is composed of the collection of its vertices. A hierarchical segmentation of a graph is a fine to coarse sequence of segmentations of this graph such that each subgraph of a segmentation is included in a subgraph of the subsequent segmentation (nesting property).

By construction, a hierarchical segmentation of a graph is parameterised by a non-negative real number indicating the level of a given segmentation in the hierarchy. At the bottom level, this number is equal to zero and each node corresponds to a subgraph so that the finest possible segmentation is obtained. At the top level, one finds the input graph. Given any two vertices, it is possible to determine the minimum level value for which these two vertices belong to the same subgraph. A key property of hierarchical segmentation is that the

[1] Standard definitions of graph theory can be found for example in the online book [10].

function that measures this minimum level is an *ultrametric*. An ultrametric is a measurement that satisfies all properties of a metric (distance) plus a condition stronger than the triangle inequality and called ultrametric inequality. It states that the distance between two vertices is lower than or equal to the maximum of the distances calculated from (i) the first vertex to an arbitrary third vertex and (ii) this third vertex to the second vertex. Denoting by d the ultrametric function and u, v, and w respectively the first, second and third vertices, the ultrametric inequality corresponds to the following equation:

$$d(u,v) \leq \max\{d(u,w), d(w,v)\}.$$

The ultrametric property of hierarchical clustering was discovered in [12]. This property was referred to in the context of morphological hierarchical segmentation for the first time in [13].

The segmentation of a graph $X = (V, E)$ relies on a function associating its edges with a weight. This function can be viewed as a measure of the degree of dissimilarity between adjacent vertices. A dissimilarity measurement [14, pp. 5–6] between the elements of a set V is a function $d^\star$ from $V \times V$ to the set of nonnegative real numbers satisfying the three following conditions: (i) $d^\star(x,y) \geq 0$ for all $x, y \in V$ (i.e., positiveness), (ii) $d^\star(x,x) = 0$ for all $x \in X$, and (iii) $d^\star(x,y) = d^\star(y,x)$ for all $x, y \in V$ (i.e., symmetry). Starting from an arbitrary dissimilarity measurement, it is possible to construct a hierarchical segmentation with the ultrametric distance between any two vertices (or subgraphs) being defined as the dissimilarity threshold level from which these two vertices (or subgraphs) belong to the same subgraph. In practice, this is achieved by an iterative procedure linking first the pair of adjacent vertices with the smallest dissimilarity value so as to form a first non-trivial subgraph (i.e., non reduced to one node). To proceed, the dissimilarity measurement between adjacent vertices needs to be extended so as to be applicable to subgraphs. Let X_i and X_j denote two subgraphs obtained at a given iteration level. The dissimilarity between between these two subgraphs is naturally defined as a function of the dissimilarities associated with the edges of X linking these two subgraphs:

$$d^\star(X_i, X_j) = \Phi\big\{d^\star(v_i, v_j) \mid v_i \in X_i, v_j \in X_j, \text{ and } \{v_i, v_j\} \in \mathcal{E}(X)\big\}.$$

Typical choices for the function Φ are the minimum, average, or maximum. In contrast to the maximum and average functions, the minimum rule (also called nearest neighbour) is insensitive to ties so that the result is unique (order independent). Segmentation by the minimum rule is referred to as single linkage segmentation. Indeed, only the pair (link) with the smallest dissimilarity value is playing a role.

In practice, there is a one to one correspondence between the objects under study and the vertices of corresponding graph. If these objects have a spatial location like the pixels of a digital image, the edges of the graph do not usually contain every possible pair of vertices but only those pairs that are spatially related according to some predefined rule. In this paper, for all examples referring to discrete images, we assume that the 4-connected graph is considered.

2.2 Constrained Connectivity

Constrained connectivity [6] extracts maximal connected components of a hierarchical representation satisfying a series of constraints expressed in terms of logical predicates. The hierarchy relies on a dissimilarity measurement originally defined as the absolute difference between pixel intensity values (i.e., local range measurement α) and is directly applicable to other dissimilarity measurements (see example in [15]).

Let (V, E) be a graph with vertex set V valued by a function f and edge set E valued by a dissimilarity function $d^\star$. The $\alpha^\star$-connected components of the graph (V, E) are the connected component of the graph (V, E') where the edge set E' is obtained by removing from the edge set E all edges which have a dissimilarity value strictly greater than the value of the non-negative parameter $\alpha^\star$: $E' = \{e_i \in E \mid d^\star(e_i) \leq \alpha^\star\}$. An equivalent definition can be obtained in terms of $\alpha^\star$-connectivity. A $\alpha^\star$-path is a sequence of vertices such that from each vertex there exists an edge to the next vertex in the sequence and with a dissimilarity (weight) less than or equal to $\alpha^\star$. Two vertices p and q are $\alpha^\star$-connected if there exists at least one $\alpha^\star$-path from p to q. By definition, a pixel is said to be $\alpha^\star$-connected to itself. The vertex set of the $\alpha^\star$-connected component of a pixel p, denoted by $\alpha^\star$-CC(p), is defined as the set of image pixels that are $\alpha^\star$-connected to p:

$$\mathcal{V}[\alpha^\star\text{-CC}(p)] = \{q \in V \mid q \text{ is } \alpha^\star\text{-connected to } p\}.$$

The edge set of $\alpha^\star$-CC(p) is defined as the set of edges of the graph induced by the vertices of $\alpha^\star$-CC(p) and whose associated dissimilarity is less than or equal to $\alpha^\star$:

$$\mathcal{E}[\alpha^\star\text{-CC}(p)] = \Big\{\{v_i, v_j\} \in \mathcal{E}\Big[X\big[\mathcal{V}[\alpha^\star\text{-CC}(p)]\big]\Big] \mid d^\star(v_i, v_j) \leq \alpha^\star\Big\}.$$

Note that the $\alpha^\star$-connectivity relation is equivalent to the single linkage clustering rule with its corresponding hierarchical representation and ultrametric [16] that can be represented by the so-called $\alpha^\star$-tree [17]. This hierarchy can be expressed in terms of a hierarchy of watersheds where the flooding starts from the image 0-dissimilar connected components [15].

Assume a series of $n \geq 1$ predicates P_k returning true on every 0-dissimilar connected component: $P_k(0^\star\text{-CC}(p)) = \text{true}$ for all pixels p. The constrained connected component of a pixel p associated with these predicates is denoted by $(P_1, \ldots, P_n)$-CC(p) and is defined as follows [9,18]:

$$\begin{aligned}(P_1, \ldots, P_n)\text{-CC}(p) = \bigvee \Big\{\alpha_i^\star\text{-CC}(p) \;\Big|\; & \\ P_k\Big(\alpha_i^\star\text{-CC}(p)\Big) &= \text{true for all } k \in \{1, \ldots, n\} \text{ and} \\ P_k\Big(\alpha_j^\star\text{-CC}(q)\Big) &= \text{true for all } j \leq i \text{ and all } q \in \alpha_i^\star\text{-CC}(p)\Big\}.\end{aligned} \quad (1)$$

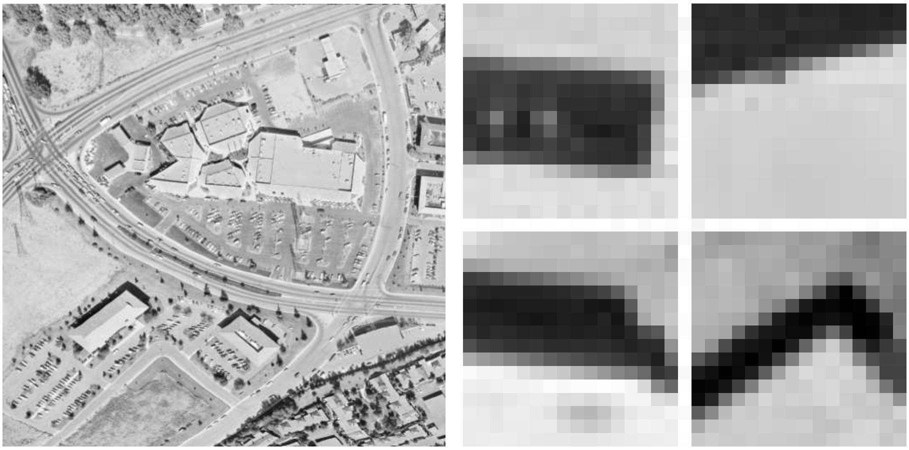
Throughout this paper, the notation α refers explicitly to the threshold value corresponding to the dissimilarity measure defined as the absolute difference while the notation $\alpha^\star$ refers to the threshold level associated with an arbitrary dissimilarity measure $d^\star$. In the next section, the notions of increasing, α-increasing, and strictly increasing paths will be needed. An increasing path of a graph with real valued vertices is a sequence of adjacent vertices whose values are monotonic increasing. An α-increasing path is an increasing path such that the maximum absolute difference between all pairs of successive pixels of the paths is equal to α. A path is strictly increasing if the values of its vertices are strictly increasing. Finally, recall that the length of a path is defined as the number of its edges, or, equivalently, the number of its vertices minus 1.

3 Characterisation of Transitions

Transitions are analysed hereafter by studying their impact on the paths linking the regions they separate while minimising the maximum absolute grey level difference between successive points of these paths.

Transitions originate from the digitisation and quantisation processes. Indeed, when sampling a signal such as an ideal 1-dimensional step edge, aliasing is usually mitigated by applying a low-pass anti-aliasing filter such as a box-filter before sampling. Then, if the resulting signal is sampled at the minimum rate required by the sampling theorem, the resulting signal will show a transition (blurred edge) with a width at least equal to the sampling distance [19]. In the case of 2-dimensional signals (images), if there is a one-to-one correspondence, both in shape and in spacing, between sensor elements and pixels, and if the sensor elements show uniform sensitivity and linear response, then the pixel value will be proportional to the mean value of the signal portion which is being covered by that pixel. Consider for example two regions separated by an ideal step edge whose orientation is expressed by an irreducible fraction b/a while including the forms $0/1$ and $1/0$ for referring to horizontal and vertical lines respectively. After digitisation, the internal boundary of the two resulting regional extrema regions will appear as a discrete Bresenham line of slope b/a. The transition is located in between these extrema regions. While its width in the direction perpendicular to that of the edge is close to 1 pixel (if sampling occurs at the minimum rate required by the sampling theorem), the least upper bound of the length of the strictly increasing (or strictly decreasing) paths linking the two extrema regions is equal to $\max(|a|, |b|)) + 1$ pixels, assuming that the intensity values are not quantised. Therefore, the theoretical greatest lower bound for the value of α of a α-increasing path linking two regions separated by an ideal step edge of arbitrary orientation, is only limited by the intensity quantisation step (assumed to be uniform). Expressing the orientation of the edge in terms of an irreducible fraction b/a and denoting by h and $\varDelta$ the intensity difference between the regions and the quantisation step respectively, the greatest lower bound of α is equal to $\max\{h/(\max(|a|, |b|) + 1), \varDelta\}$.

This effect can be observed in practice as shown in Fig. 2 on a series of 4 zooms of a typical aerial image. Each zoom displays the transitions between dark and bright regions (the former are shadows cast by buildings and the latter



(a) 512 × 512 aerial image.

(b) Four 16×16 samples of *(a)* with clearly visible transitions.

Fig. 2. A grey level image and a series of 4 zooms with transitions at the interface between bright and dark regions. The input image originates from the miscellaneous section of the USC-SIPI image database, see http://sipi.usc.edu/database/

the buildings themselves or the ground). In these zooms, notice that the length of the longest strictly increasing paths linking dark to bright regions increases when the slope of the edge between these regions gets closer to the horizontal or vertical directions. Accordingly, the α value of the corresponding α-increasing paths decreases. Experimental results on controlled and real data with quantitative measurements confirm these observations. The previous examples also show that partitions relying on α-connectivity will depend on the orientation of the image objects. Consequently, the use of subsequent constraints applied to the resulting α-connected components cannot suppress this dependency.

4 Handling Transitions in Constrained Connectivity

As stated in the introduction, constrained connectivity based on α-connectivity is effective in preventing chaining through transitions because the resulting connected components would violate the input constraints, but it may generates many small connected components between adjacent larger connected components. In addition, if none of the α-connected components of an α-hierarchy is matching a given homogeneous region because its internal homogeneity is higher than its external isolation, the use of additional constraints does not help. While appropriate post-processing can address the former problem, this approach cannot solve the latter problem. Both problems can be handled simultaneously by pre-filtering the image as shown in section 4.1. An alternative approach is to consider a contrast-based dissimilarity measurement that does not allow chaining through transitions as proposed in section 4.2.

4.1 Pre-filtering

Pre-filtering consists in pre-processing the image in such a way that transition regions are suppressed or at least reduced. This can be achieved by most neighbourhood-based contrast enhancement techniques. For example, iterations of the Kramer-Bruckner enhancement [20] (further developed in mathematical morphology in [21,22]) leads to images containing only local minima and maxima and therefore no more transition pixels. However, this approach modifies drastically the image and has side-effects such as the creation of halos [21].

By its very nature, a transition pixel is a pixel that is not a local extremum [18]. The transition pixels of a grey tone image f correspond to the non-zero values of the point-wise minimum between the elementary gradients by erosion and dilation of f (denoted by ρ^{ε} and ρ^{δ} respectively):

$$p, \text{ transition pixel of } f \Leftrightarrow \left[\rho^{\varepsilon}(f) \wedge \rho^{\delta}(f)\right](p) \neq 0. \tag{2}$$

Pre-filtering consists in preprocessing the image so as to reduce the occurrence of transition pixels. This can be achieved by applying the following contrast enhancement filter. First, the local extrema of the input image are extracted. Then, they are expanded to generate the local extrema mosaic using a region growing process driven by a similarity measurement [23]. Each region of the local extrema mosaic is then set to the value of the local extremum that generated this region. This leads to a contrast enhanced image with only few remaining transition pixels. Consequently, constrained connected components of the enhanced image are less likely to chain from one component to another so that chaining within homogeneous regions is favoured while being prevented through transitions. Note that this method inherits of the order dependence of most region growing algorithms. The iterative ramp sharpening method proposed in [24] is order independent and avoid ringing (halos) but it is not as effective as the extrema mosaic for mitigating the occurrence of transition pixels in the sense of Eq. 2.

4.2 Contrast-Based Dissimilarity Measurement

To prevent chaining through transition pixels, an estimation of the minimum α-dissimilarity (i.e., absolute difference of the intensity of adjacent pixels) along the most contrasted increasing paths[2] of three pixels going through the considered pixel is needed. The most contrasted path going through a transition pixel links its (or one of its) adjacent pixel(s) having the lowest intensity level to its (or one of its) adjacent pixels having highest intensity level. Therefore, the contrast of the most contrasted increasing paths of an arbitrary pixel is equal to the value of the morphological gradient ρ at this pixel if it is a transition pixel, 0 otherwise. The resulting map is called transition map in [18]. Accordingly, the minimum α-dissimilarity along the most contrasted paths going through a pixel is equal to

[2] The contrast of an increasing path is defined as the absolute difference of the intensity values of its end points.

the point-wise *minimum* value of the values of the gradients by erosion ρ^{ε} and dilation ρ^{δ} at this pixel. We call the corresponding operator the *inf-increment* operator and denote it by $\rho^{\wedge}$:

$$\rho^{\wedge} = \rho^{\varepsilon} \wedge \rho^{\delta}.$$

Alternatively, if transition paths need to be restricted to horizontal or vertical sequences of pixels, the inf-increment operator should be redefined accordingly:

$$\rho^{\wedge} = \bigvee_{\theta \in \{0, \pi/2\}} \{\rho^{\varepsilon}_{\theta} \wedge \rho^{\delta}_{\theta}\},$$

where $\rho^{\varepsilon}_{\theta} = \mathsf{id} - \varepsilon_{\theta}$ and $\rho^{\delta}_{\theta} = \delta_{\theta} - \mathsf{id}$ with ε_{θ} (resp. δ_{θ}) denoting the elementary directional erosion (resp. dilation) in the direction θ.

The extension of these definitions to larger neighbourhoods is left for an extension of this paper. Because gradients by erosion and dilation boil down to local downward and upward contrast measurements (grey level differences), they can be combined with absolute grey level differences when designing a contrast based dissimilarity measure. Since the goal is to prevent chaining through transitions, the combination should be based on the point-wise maximum rule. We obtain therefore the following dissimilarity measure:

$$d^{\star}(p, q) = \max\big\{|f(p) - f(q)|, \rho^{\wedge}(p), \rho^{\wedge}(q)\big\}. \tag{3}$$

With this dissimilarity value, pixels located in the middle of the paths linking two regions by chaining through transitions (see Sec. 3) have a dissimilarity value close to half of the absolute difference of the values of the regions they link. Therefore, this dissimilarity measure prevents linking through transitions while favouring it through homogeneous regions.

5 Results and Discussion

Results are illustrated on a synthetic image displayed in Fig. 3a. This image consists of two regions (with intensity values set to 1 or 0 and 5 or 6 respectively) separated by a one pixel thick vertical ramp with steps of one intensity level. It follows that there are only two types of α-connected components for this image: either the pixels themselves (the 0-CCs) or only one connected component covering the whole image definition domain (the 1-CC). Assuming 4-adjacency, the transition between these regions goes by steps of one intensity level. The contrast enhanced image according to the method of Sec. 4.1 is shown in Fig. 3b. Note that the value of the middle pixel of the transition region is to either 6 or 0 (here 0), the choice depending on the implementation. Two α-connected components matching the left and right regions is obtained for all values of α in the range $[1, 4]$ as shown in Fig. 3c.

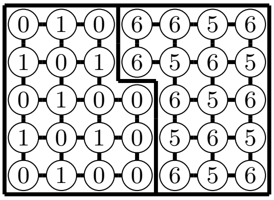
Figures 4a–e show the results obtained with contrast-based dissimilarity measurement and assuming also 4-adjacency. In this case, no pre-filtering is required

0	1	0	5	6	5	6
1	0	1	4	5	6	5
0	1	0	3	6	5	6
1	0	1	2	5	6	5
0	1	0	1	6	5	6

(a) Input synthetic image

0	1	0	6	6	5	6
1	0	1	6	5	6	5
0	1	0	0	6	5	6
1	0	1	0	5	6	5
0	1	0	0	6	5	6

(b) Contrast enhanced image



(c) α-CCs for $\alpha \in [1, 4]$

Fig. 3. Handling transitions using pre-filtering method

0	1	0	5	1	0	1
1	0	1	3	1	1	0
0	1	0	3	3	0	1
1	0	1	1	3	1	0
0	1	0	1	5	0	1

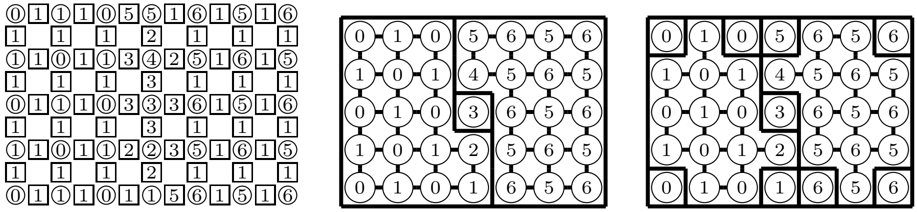
(a) Internal gradient ρ^{ε}

1	0	5	1	0	1	0
0	1	3	1	1	0	1
1	0	3	3	0	1	0
0	1	1	3	1	0	1
1	0	1	5	0	1	0

(b) External gradient ρ^{δ}

0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	0	0	3	0	0	0
0	0	1	1	1	0	0
0	0	0	1	0	0	0

(c) Inf-increment $\rho^{\wedge} = \rho^{\varepsilon} \wedge \rho^{\delta}$



(d) Dissimilarity map $d^{\star}$ (boxed values) following Eq. 3

(e) $2^{\star}$-CCs

(f) For comparison: using degree-based dissimilarity

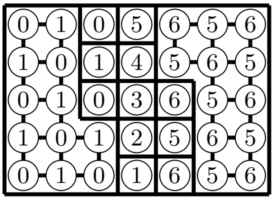
Fig. 4. Handling transitions using modified dissimilarity measures. (a)–(e) The proposed contrast-based dissimilarity. (f) The degree-based dissimilarity of [15]. The input image is shown in Fig. 3a.

and the resulting segmentation (Fig. 4e) is uniquely defined (no ties). For comparison, Fig. 4f shows the output of a degree-based dissimilarity [15] (with $\alpha = 2$ and $\deg \geq 3$) preventing chaining through transitions but creating one pixel CCs for all pixels having less than three α-neighbours (i.e., $\deg < 3$), e.g., all corners.

Note that chaining through transitions does not arise in the smooth connection proposed in [5]. Indeed, contrary to dissimilarity based connectivity, smooth connection is not based on graph adjacency but on a connective criterion defined in terms of balls on which the function is Lipschitz so that any smooth linkage between two regions must necessarily have a minimum width. However, the drawback of this approach is that flat and thin bright and dark image structures are also decomposed into isolated pixels (singletons of the partition) as well as all pixels the edges of the smooth regions. More precisely, the support of the singleton connected components of the smooth connection with Lipschitz constant set to k are obtained by thresholding the point-wise maximum of the elementary gradients by erosion and dilations for all values that are greater than or equal

1	1	5	5	1	1	1
1	1	3	3	1	1	1
1	1	3	3	3	1	1
1	1	1	3	3	1	1
1	1	1	5	5	1	1

(a) Sup-increment $\rho^{\vee} = \rho^{\varepsilon} \vee \rho^{\delta}$



(b) Smooth connection

Fig. 5. Smooth connection. (a) Sup-increment function of the synthetic image shown in Fig. 3a. (b) Resulting connected regions according to the smooth connection [5] with Lipschitz constant k set to 1.

to k. The point-wise maximum of the elementary gradients by erosion and dilations is called the increment function in [5]. By analogy with the inf-increment operator put forward in this paper, we call the operator underlying the increment function of [5] the sup-increment operator and denote the it by $\rho^{\vee}$, i.e., $\rho^{\vee} = \rho^{\varepsilon} \vee \rho^{\delta}$. For example, the output of the sup-increment operator of Fig. 3a is displayed in Fig. 5a (see Figs. 4a–b for the gradients by erosion and dilation) and the output of the corresponding smooth connection is shown in Fig. 5b.

6 Conclusion

Having all pixels in focus and sampling close to the minimum rate is essential when considering constrained connectivity. Fortunately, these conditions are fulfilled for most aerial and satellite images. If not, transitions in the direction perpendicular to the object edge could span over many pixels so that the proposed dissimilarity measure would not be as effective for blocking chaining through these transitions. In this latter case, other dissimilarities would need to be developed. For example, the concept of frequent and dependent connectivity in the framework of decision based connectivity [25] could be used to mitigate chaining through transitions as briefly hinted in the latter paper. A paper unifying these concepts and assessing quantitatively their impact on the constrained connectivity partitioning of satellite images is in preparation.

References

1. Nagao, M., Matsuyama, T., Ikeda, Y.: Region extraction and shape analysis in aerial photographs. Computer Graphics and Image Processing 10(3), 195–223 (1979)
2. Baraldi, A., Parmiggiani, F.: Single linkage region growing algorithms based on the vector degree of match. IEEE Transactions on Geoscience and Remote Sensing 34(1), 137–148 (1996)
3. Meyer, F., Maragos, P.: Morphological scale-space representation with levelings. In: Nielsen, M., Johansen, P., Fogh Olsen, O., Weickert, J. (eds.) Scale-Space 1999. LNCS, vol. 1682, pp. 187–198. Springer, Heidelberg (1999)

4. Zanoguera, F., Meyer, F.: On the implementation of non-separable vector levelings. In: Talbot, H., Beare, R. (eds.) Proc. of VIth ISMM, Sydney, CSIRO, pp. 369–377 (2002)
5. Serra, J.: A lattice approach to image segmentation. Journal of Mathematical Imaging and Vision 24(1), 83–130 (2006)
6. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(7), 1132–1145 (2008)
7. Gower, J., Ross, G.: Minimum spanning trees and single linkage cluster analysis. Applied Statistics 18(1), 54–64 (1969)
8. Wishart, D.: Mode analysis: a generalization of nearest neighour which reduced chain effect. In: Cole, A. (ed.) Numerical Taxonomy, pp. 282–311. Academic Press, New York (1968)
9. Soille, P.: On genuine connectivity relations based on logical predicates. In: Proc. of 14th ICIAP, Modena, Italy, pp. 487–492. IEEE, Los Alamitos (2007)
10. Diestel, R.: Graph Theory, 4th edn. Graduate Texts in Mathematics. Springer, Heidelberg (2010)
11. Zucker, S.: Region growing: childhood and adolescence. Computer Graphics and Image Processing 5, 382–399 (1976)
12. Johnson, S.: Hierarchical clustering schemes. Psychometrika 32(3), 241–254 (1967)
13. Meyer, F.: An overview of morphological segmentation. International Journal of Pattern Recognition and Artificial Intelligence 15(7), 1089–1118 (2001)
14. Jardine, N., Sibson, R.: Mathematical Taxonomy. Wiley, London (1971)
15. Najman, L., Soille, P.: On morphological hierarchical representations for image processing and spatial data clustering. In: Proc. of WADGMM ICPR Workshop, Istanbul, IAPR, pp. 52–61 (August 2010)
16. Soille, P.: On range based connectivity relations for image simplification and segmentation. Technical report, European Commission, Joint Research Centre (2006), http://www.jrc.ec.europa.eu
17. Ouzounis, G., Soille, P.: Pattern spectra from partition pyramids and hierarchies. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 108–119. Springer, Heidelberg (2011)
18. Soille, P., Grazzini, J.: Constrained connectivity and transition regions. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 59–69. Springer, Heidelberg (2009)
19. Gerbrands, J.: Segmentation of noisy images. PhD thesis, TU Delft, The Netherlands (1988)
20. Kramer, H., Bruckner, J.: Iterations of non-linear transformations for enhancement on digital images. Pattern Recognition 7, 53–58 (1975)
21. Meyer, F., Serra, J.: Contrasts and activity lattice. Signal Processing 16(4), 303–317 (1989)
22. Schavemaker, J., Reinders, M., Gerbrands, J., Backer, E.: Image sharpening by morphological filtering. Pattern Recognition 33, 997–1012 (2000)
23. Soille, P.: Constrained connectivity for the processing of very high resolution satellite images. International Journal of Remote Sensing 31(22), 5879–5893 (2010)
24. Grazzini, J., Soille, P.: Iterative ramp sharpening for structure/signature-preserving simplification of images. In: Proc. of 20th Int. Conf. on Pattern Recognition, Istanbul, pp. 4585–4589. IEEE, Los Alamitos (2010)
25. Gueguen, L., Soille, P.: Frequent and dependent connectivities. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 120–131. Springer, Heidelberg (2011)

Pattern Spectra from Partition Pyramids and Hierarchies

Georgios K. Ouzounis and Pierre Soille

Geo-Spatial Information Analysis for Global Security and Stability,
Global Security and Crisis Management Unit,
Institute for the Protection and Security of the Citizen,
Joint Research Centre, European Commission,
Via E. Fermi 2749, I-21027 Ispra (VA), Italy
{georgios.ouzounis,pierre.soille}@jrc.ec.europa.eu
http://isferea.jrc.ec.europa.eu/

Abstract. Constrained connectivity relations partition the image definition domain into connected sets of maximal extent. The homogeneity and maximality of the resulting cells is subject to non-connective criteria that associate to logical predicates. The latter are based on attribute metrics, and are used to counter the leakage effect of the single-linkage clustering rule. Linkage is controlled by some dissimilarity measure and if unconstrained, may be used to generate regular connectivity classes. In this paper we introduce a hierarchical partition representation structure to map the evolution of components along the dissimilarity range in the absence of constraints. By contrast to earlier approaches, constraints may be put in place on the actual structure in the form of filters. This allows for custom and interactive segmentation of the image. Moreover, given an instance of the dissimilarity measure, one can retrieve all connected sets making up the corresponding connectivity class, directly from the hierarchy. The evolution of linkage relations with respect to the attributes on which the predicates are based on is used to compute a new type of pattern spectrum that is demonstrated on two real applications.

1 Introduction

Image segmentation in the context of connected morphology [18] is the partitioning of the image definition domain into homogeneous cells of maximal extent. Maximality is guaranteed through a connective criterion of homogeneity, i.e. one that generates a connection [17, Chap. 2] from the homogeneous image regions. Connective criteria however, introduce leakage effects on the connected components and examples are discussed in [12].

Constrained connectivity [19] provides a segmentation framework that relies on a sub-connection of the canonical path-based graph connectivity. This is referred to as *dissimilarity-based* or α-connectivity [19] and can be constrained through a series of logical predicates. Utilizing appropriate constraints minimizes the leakage effect in the process of generating homogeneous image regions by

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 108–119, 2011.

preventing the creation of a maximal partition [18]. That is because the binary criterion associated to each predicate is not connective. This works in the benefit of many applications since it allows for strict controls over the segmentation process.

The α-connected components (α-$\texttt{CCs}$) [19], also known as single-linkage components [7], are equivalence classes on the image definition domain, thus defining a unique partition. Constrained α-partition cells are non maximal in the sense of a connective criterion, and an example is discussed in Section 3. α-partitions however, irrespective of any constraints, are totally ordered with respect to the dissimilarity range [15] and can be organized in a hierarchical structure that we refer to as a *partition pyramid*. This can be further reduced to a compact equivalent, the *partition hierarchy*, which is characterized by strict inclusion, thus resolving redundancies caused by replicating components over multiple pyramid levels. Both structures are discussed in Section 4. Running queries based on cell attributes [2] allows for customizing the segmentation in ways not supported by the conventional connective segmentation scheme. The work in this paper deals with computing pattern spectra [5] from both types of partition representation structures. This is described in Section 5 and it is followed by two applications given in Section 6. A short discussion and conclusions are given in Section 7.

2 Preliminaries

Let I be a gray-tone image and E be its definition domain, i.e. a Euclidean subspace. A partition $\mathbf{P}$ of E is its division into a set of non overlapping and non-empty *cells*, the union of which is equal to E. The cells of $\mathbf{P}$ are both collectively exhaustive and mutually exclusive with respect to the set being partitioned. The formal definition as given in [17, Chap. 1], is the following:

Definition 1. *Let E be the definition domain of an image. A partition $\mathbf{P}$ of E is a mapping $x \rightarrow \mathbf{P}(x)$ from E into the power set of E, denoted by $\mathcal{P}(E)$, such that:*

1. $\forall x \in E \Rightarrow x \in \mathbf{P}(x)$;
2. $\forall x, y \in E \Rightarrow \mathbf{P}(x) = \mathbf{P}(y)$ *or* $\mathbf{P}(x) \cap \mathbf{P}(y) = \emptyset$.

The term $\mathbf{P}(x)$ above indicates a cell of $\mathbf{P}$ marked by/containing a point $x \in E$. It follows that:

$$\bigcup_{x \in E} \mathbf{P}(x) = E. \tag{1}$$

A partitioning scheme frequently encountered in image analysis is the separation of the image content into foreground and background components. This dichotomy can be realized by connected operators, examples of which are the connectivity openings and closings. Given a point $x \in E$ that marks a set $X \subseteq E$, the connectivity opening Γ_x extracts the set of maximal extent containing x or returns $\emptyset$ otherwise. A set $C \subseteq E$ is of maximal extent if there is no other set $C' \supset C$ such that $C' \subseteq E$ and $C' \in \mathcal{C}$. In this case C is referred to as a *connected component* according to the connectivity class $\mathcal{C}$.

A connectivity class is the family of all the sets of a space E that are connected according to some notion of connectivity. The axiomatic definition given in [17, Chap. 2] states:

Definition 2. *Let E be an arbitrary non-empty space. A* connectivity class *or* connection $\mathcal{C}$ *is any family in $\mathcal{P}(E)$ such that:*

1. $\emptyset \in \mathcal{C}$;
2. $\forall\, x \in E, \{x\} \in \mathcal{C}$;
3. *for each family $\{C_i, i \in L\} \subseteq \mathcal{C}$, $\bigcap_i C_i \neq \emptyset$ implies $\bigcup_i C_i \in \mathcal{C}$, where L is an index set.*

Connectivity openings are directly related to connectivity classes [17, Chap. 2], and can be customized to address more general notions of connectivity such as clustering, contraction [14,1], mask-based connectivity [11] or partition-induced connectivity [18,12]. They are anti-extensive, increasing and idempotent operators and form the basis of attribute filters [2]. The latter are edge preserving operators that extract connected components which satisfy some attribute criterion. Attribute filters and other connected operators can be organized in families known as *granulometries* [17], with respect to some scale or attribute parameter s from a totally ordered set, from which one can study the distribution of image detail with respect to the concerned attribute.

Let $S = \{0, s_1, ..., s_{\max}\}$, be a set of totally ordered thresholds for some increasing attribute. Formally, a granulometry of a binary image X can be defined as a decreasing family of attribute openings $\{\Gamma^s \mid s \in S\}$ for which:

$$\forall s, s' \in S \Rightarrow \Gamma^s(\Gamma^{s'}(X)) = \Gamma^{s'}(\Gamma^s(X)) = \Gamma^{\max(s,s')}(X). \tag{2}$$

Granulometries for non-increasing operators have been investigated in [22]. The order in that case is preserved by utilizing appropriate filtering rules. The distribution of image detail that is often given by the sum of pixels, with respect to one or more attributes is a histogram that is referred to as the granulometric curve or *pattern spectrum* [5]. An attribute class or histogram bin to which a point $x \in X$ contributes, is the smallest value of s for which $x \notin \Gamma^s(X)$. The pattern spectrum by $\mathtt{PS}_{\Gamma^s}(X)$ applying an attribute-specific granulometry $\{\Gamma^s\}$ to a binary image X is defined as [22]:

$$(\mathtt{PS}_{\Gamma^s}(X))(u) = -\left.\frac{\mathrm{d}\xi(\Gamma^s(X))}{\mathrm{d}s}\right|_{s=u}. \tag{3}$$

The term ξ is the Lebesgue measure, which for $n = 2$ is the set area.

3 Constrained Connectivity

Consider a gray-tone image I projected on a graph space in which, vertices (nodes) correspond to atomic elements, and edges to pairs of adjacent vertices. A path $\{x \rightsquigarrow y\}$ between any two elements x to y is a chain of pairwise adjacent elements commonly given in the form of $\{x \rightsquigarrow y\} \equiv \langle x_0 = x, x_1, ..., x_n = y\rangle$.

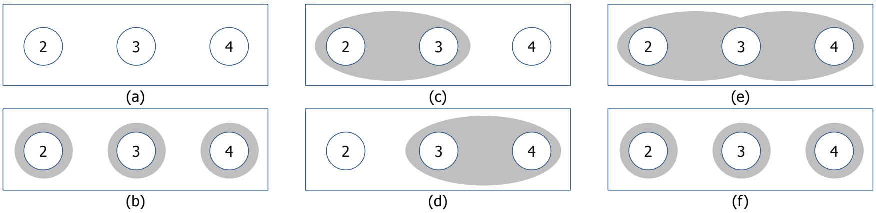


Fig. 1. Three points on the input space labeled by some attribute value, (a). Setting α=1 yields 6 connected sets; the 3 singletons in (b), the set $\{2, 3\}$ in (c), the set $\{3, 4\}$ in (d), and the set of maximal extent, i.e. α-connected component, in (e). For the last three, the difference between neighbors is no greater than 1. The total range constraint $\omega = \alpha$ yields three singleton sets since the ω range of the 1-CC in (e) is 2, i.e. ω prevents its creation, (f).

Links between adjacent nodes are called edges. Assuming that d is the slope, i.e. the intensity difference, between any two adjacent elements, then setting $\mathtt{d} = 0$ leads to the definition of *flat zones* [16], which are connected image regions of constant intensity. Note that d is not necessary intensity-specific. The term flat-zone refers to components with members of 0 dissimilarity between them, and in the more general case, single-linkage [7] or α-connected components [19], also known as *quasi-flat zones* [6], can be defined as follows:

$$C^{\alpha}(x) = \{x\} \cup \{y \mid \exists\{x \rightsquigarrow y\} : \forall x_i \in \{x \rightsquigarrow y\} \ \wedge \ x_i \neq y,\ \mathtt{d}(x_i, x_{i+1}) \leq \alpha\}. \quad (4)$$

That is, all atomic elements are connected sets themselves, and a connected set of maximal extent, i.e. a connected component, marked by a point $x \in E$ is the union of all points $y \in E$, such that for each one there exists a path to x in which all adjacent elements have a dissimilarity measure less than or equal to α. The α parameter in the above definition is a threshold; if the difference between some attribute of x and y is less than or equal to α, the two are directly connected, i.e. there exists an edge between x and y, thus are members of the same α-connected component $C^{\alpha}(x)$. The case in which $\mathtt{d}(x, y) > \alpha$ does not imply that x and y do not belong to the same $C^{\alpha}(x)$ but only that there is no direct linkage between them. Note that any α-CC given by (4), satisfies the two last conditions of Def. 2, i.e. α-connectivity is a sub-connection of the canonical path-wise connection on a graph space. Fig. 1 shows an example of a single 1-CC consisting of three points. In image (e) the union of two α-connected sets $\{2, 3\}$ and $\{3, 4\}$ that have a non-empty intersection, is α-connected too.

Connectivity relations based on dissimilarity measures are known to suffer from leakage effects through paths in which adjacent elements differ less than α. One solution, proposed explicitly for this problem, is the introduction of constraints through a sequence of logical predicates based on various attributes [21]. An example is the ω range [19], i.e. the total intensity variation allowed within a component, which leads to the following definition:

$$C^{(\alpha,\omega)}(x) = \max\left\{C^{\alpha_j}(x) \mid \alpha_j \leq \alpha \text{ and } \mathrm{R}(C^{\alpha_j}(x)) \leq \omega\right\}. \quad (5)$$

A $C^{(\alpha,\omega)}(x)$ connected component is essentially the maximal indexed $C^{\alpha_j}(x)$ containing the point $x \in E$ that does not exceed the local and global dissimilarity threshold α and ω respectively. The latter is given by the R() functional.

The two conditions of (5) can be expressed as Boolean valued functions, each returning **true** when the associated argument satisfies the corresponding predicate and **false** otherwise. Consider a binary criterion σ [18] enumerating the logical output of each predicate in the general case of constrained connectivity.

Definition 3. *Let E and T be two arbitrary sets and let $\mathcal{F}$ be a family of functions from E into T. A criterion σ on the class $\mathcal{F}$ is a binary function from $\mathcal{F} \times \mathcal{P}(E)$ into $\{0,1\}$ such that for each function $f \in \mathcal{F}$, and for each set $X \in \mathcal{P}(E)$:*

1. *$\sigma[f,X] = 1$, if the predicate returns **true**,*

2. *$\sigma[f,X] = 0$, if the predicate returns **false**.*

Moreover, it is assumed that for all functions the respective criteria are satisfied on $\emptyset$, i.e.:

$$\sigma[f,\emptyset] = 1, \forall\, f \in \mathcal{F}. \tag{6}$$

In the interest of segmentation as described in [18], searching for the largest partition in conjunction to the connectivity class axiomatics, Serra concludes to the more explicit *connective criterion* which is defined as follows:

Definition 4. *A criterion $\sigma : \mathcal{F} \times \mathcal{P}(E) \rightarrow \{0,1\}$ is connective if for each set $f \in \mathcal{F}$ the sets X for which the predicate(s) is (are) satisfied, generate a connection, i.e.:*

1. *for the class of singletons $\mathcal{S}$ and $\forall\, f \in \mathcal{F}, \{x\} \in \mathcal{S} \Rightarrow \sigma[f,\{x\}] = 1$;*
2. *$\forall\, f \in \mathcal{F}$ and $\forall \{X_i\} \in \mathcal{P}(E) : \sigma[f,\{X_i\}] = 1$ the following holds: $\bigcap X_i \neq \emptyset$ and $\bigwedge \sigma[f,\{X_i\}] = 1 \Rightarrow \sigma\Big[f, \bigcup X_i\Big] = 1$.*

Returning to the example of Fig. 1, it is seen that the three points of image (a) define 6 α-connected sets for $\alpha = 1$; that is the 3 singletons, the sets $\{2,3\}$ and $\{3,4\}$, and the 1-CC consisting of all 3 points, i.e. the connected set of maximal extent. The conditions for the criterion σ to be connective, are satisfied. In particular, the sets $\{2,3\}$ and $\{3,4\}$ have a non-empty intersection and the α predicate is satisfied in their union. This however, is not the case when a further constraint is added. Image (f) shows an example for the case that $\omega = \alpha$ (ω is the total range). The corresponding criterion σ is satisfied on each of the two overlapping connected sets but fails on their union because the total range is 2. Thus, σ does not generate a connection for $\alpha = \omega = 1$, i.e. it is not a connective criterion.

4 Partition Pyramids and Hierarchies

Creating a partition of the image definition domain based on constrained connectivity relations was so far dealt with by constraining the evolution of regular α-CCs. In this section we propose an alternative method, in which the evolution of components is computed under the absence of constraints, and represented in a hierarchical structure. Constraints are put in place in a separate stage, through binary attribute criteria on the respective α-CCs. Creating the desired partition becomes an equivalent process to regular attribute filtering, and allows for interactive segmentation.

Recall that the notion of connectivity defines an equivalence relation on the image definition domain. Like with regular adjacency-based connectivity, dissimilarity-based connectivity was shown to be reflexive, symmetric and transitive [21,19], thus leading to a partition of any given I to a finite set of cells that correspond to α-CCs. The cells of $\mathbf{P}^\alpha$ are also known as *equivalence classes.* Note that all elements in a given cell are equivalent among themselves and no element is equivalent to any other element from a different cell.

Given an image I, let $\Pi^A(E)$ be the set of all α-partitions of its definition domain E. A is called the alpha dissimilarity range and is defined as a vector of thresholds $A = [0, 1, ..., \alpha_{\max}]$ whose upper bound depends on the dissimilarity measure d. Given a point $x \in E$ marking a cell of a partition $\mathbf{P}^\alpha \in \Pi^A(E)$, with $\alpha \in A$ and assuming that $|A| > 1$, then for any other $\alpha' \in A$:

$$\forall x \in E, \text{if } \alpha' < \alpha \Rightarrow C^{\alpha'}(x) \subseteq C^\alpha(x) \Rightarrow \mathbf{P}^{\alpha'} \preccurlyeq \mathbf{P}^\alpha. \tag{7}$$

The relation $\preccurlyeq$ denotes a notion of order with respect to $\alpha \in A$ [19]. The family of ordered partitions of E for the entire α dissimilarity range is defined as follows:

Definition 5. *A partition pyramid of E assuming that $|A| > 1$, is a mapping $\triangle^A : E \rightarrow \Pi^A(E)$ given by:*

$$\triangle^A = \left\{\mathbf{P}^{\alpha=0}, \mathbf{P}^{\alpha=1}, ..., \mathbf{P}^{\alpha_{\max}}\right\} \;\mid\; \mathbf{P}^{\alpha'} \preccurlyeq \mathbf{P}^\alpha, \forall\, \alpha' < \alpha \textit{ with } \alpha', \alpha \in A. \tag{8}$$

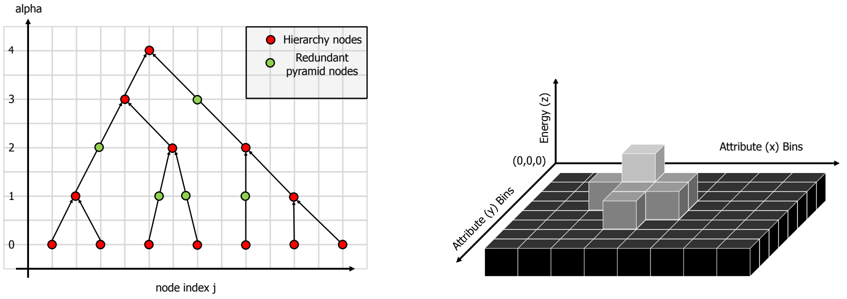
A pyramid level $\triangle^A_\alpha \in \triangle^A$ is a partition $\mathbf{P}^\alpha$ of E, with $\alpha \in A$. Note that the base of the pyramid corresponds to the finest, and the tip to the coarsest partition of E, with respect to the dissimilarity measure d.

Consider a variable $j \in J^\alpha$, in which $J^\alpha \subseteq \mathbb{Z}$ is an index set, employed to address the α-CCs making up $\mathbf{P}^\alpha$. Given a point $x \in E$:

$$\exists! j \in J^\alpha : x \in C^\alpha_j. \tag{9}$$

This complies with the first condition of Def.1, i.e. each point belongs to a unique cell of the partition, while the second condition follows, i.e. $\bigcup_{j \in J^\alpha} C^\alpha_j = E$.

A partition pyramid is a multi-scale partition representation structure which is often characterized by large redundancies. This is due to the persistence of some α-CCs in more that one level of $\triangle^A$. To counter this, an index mapping of α-CCs is introduced, that leads to a hierarchical representation configured with strict inclusion.



(a) (b)

Fig. 2. Example of a 5-level α-partition hierarchy and pyramid (a). The union of all nodes defines the pyramid structure. The red nodes alone are the resulting hierarchy. The green nodes represent the redundancy of the pyramidal representation. Example of a 2D pattern spectrum (b).

Definition 6. *Let $\triangle^A$ be an α-partition pyramid of a gray-tone image I, defined for a dissimilarity measure* $\mathtt{d}$ *over a set of threshold values A. An α-partition hierarchy $\pitchfork^A$ is a family of ordered mappings $\pitchfork^A_\alpha$: $J^\alpha \rightarrow K^\alpha$ with $K^\alpha \subseteq J^\alpha$, given by:*

$$\pitchfork^A = \left\{ \pitchfork^A_{\alpha=0}, \pitchfork^A_{\alpha=1}, ..., \pitchfork^A_{\alpha_{\max}} \right\} \mid \pitchfork^A_{\alpha'} \prec \pitchfork^A_{\alpha},\ \forall\ \alpha' < \alpha \textit{ with } \alpha', \alpha \in A, \tag{10}$$

and $\forall \alpha \in A \setminus 0$ and $\forall j \in J^\alpha$:

$$\pitchfork^A_\alpha = \left\{ C^\alpha_j \mid \left(C^\alpha_j \in \triangle^A_\alpha \right) \wedge \left(C^\alpha_j \notin \triangle^A_{\alpha-1} \right) \right\}. \tag{11}$$

In words, each level of the hierarchy $\pitchfork^A$ contains explicitly only those elements of the corresponding pyramid level, that appear for the first time.

The α-partition hierarchy $\pitchfork^A$ can be viewed as a lossless compression of an α-partition pyramid. Each level $\triangle^A_\alpha$ can be restored as follows:

$$\triangle^A_\alpha = \left\{ \bigvee_{\alpha' \in [0,\ldots,\alpha],\ j \in J^{\alpha'}} C^{\alpha'}_j \mid C^{\alpha'}_j \in \pitchfork^A \right\}, \tag{12}$$

i.e. it is the set of all maximal $C^{\alpha'} : \alpha' \leq \alpha$, which further define a partition of E. Fig. 2(a) shows an example of a five-level color-coded partition and hierarchy, to differentiate between redundant and essential (unique) nodes.

5 Pattern Spectra

Pattern spectra are typically computed from a set of operators, ordered with respect to some global variable, and an example is the morphological granulometries. α-partition pyramids by contrast to regular connected granulometries, provide an order with respect to the local dissimilarity parameter α. To compute the α-connected pattern spectra of a gray-tone image I, one needs to compute the stack of all possible α-partitions.

Given any partition $\mathbf{P}$ of E with cells indexed by a variable $j \in J$, the u entry or bin of the pattern spectrum is equivalent to the sum of the areas of all partition cells with attribute measures contained within the bounds of u, i.e.:

$$(\mathtt{PS}_{\mathbf{P}}(I))(u) = \sum_{j:\mathrm{Bin}(\mathbf{P}_j)=u} \mathrm{area}(\mathbf{P}_j), \tag{13}$$

in which "Bin" is the binning function described later, and "area" is the set area. Extending (13) for an α-partition pyramid we arrive at:

$$(\mathtt{PS}_{\triangle^A}(I))(u) = \sum_{\alpha \in A} \sum_{j \in J^\alpha:\mathrm{Bin}(C_j^\alpha)=u} \mathrm{area}(C_j^\alpha). \tag{14}$$

It can be seen that any α-$\mathtt{CC}$ that remains invariant for a limited range of dissimilarity values, is accessed in each iteration of the inner sum. This type of redundancy is typical for the pyramidal partition representation and an example is shown in Fig. 2(a). The union of all nodes corresponds to $\triangle^A$, with $A = [0, 1, 2, 3, 4]$, and the green nodes represent the redundancy.

This, in the case of an α-partition hierarchy, also referred to as the α-tree [9], is countered by introducing the multiplier term $\Delta\alpha$, also known as component *lifetime* or *persistence*.

Assume two nested α-$\mathtt{CC}$s: $C^{\alpha_p} \subset C^\alpha$ with α_p given by:

$$\alpha_p = \vee \alpha' \in A \ : \ (\alpha' < \alpha) \ \wedge \ (C_j^\alpha, C_{j'}^{\alpha'} \in \pitchfork^A). \tag{15}$$

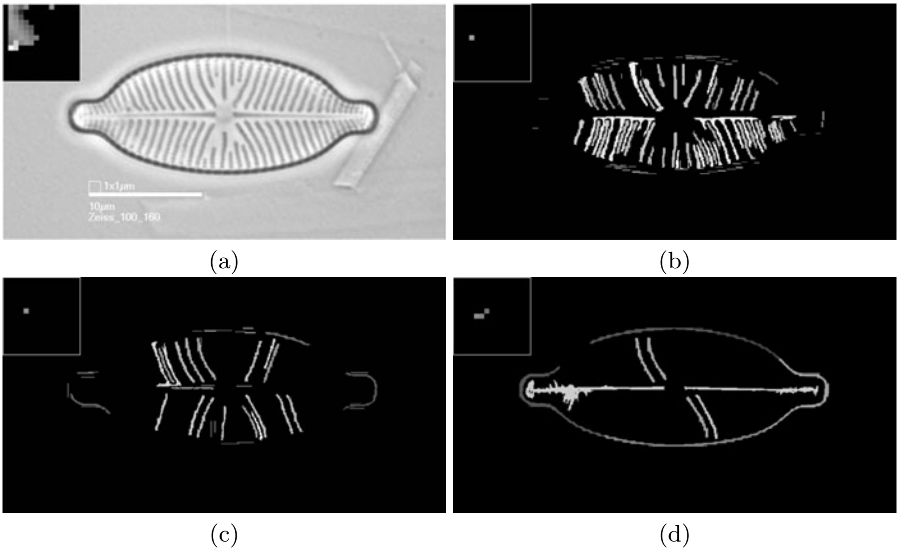
The component lifetime is defined as $\Delta\alpha = \alpha - \alpha_p$, and assuming a new index set K derived from J, with $k \in K^\alpha$, $\forall\ \alpha \in A$, the pattern spectrum computed from the α-partition hierarchy is given by:

$$(\mathtt{PS}_{\pitchfork^A}(I))(u) = \sum_{\alpha \in A \setminus 0} \sum_{k \in K^\alpha:\mathrm{Bin}(C_k^\alpha)=u} \mathrm{area}(C_k^\alpha)\Delta\alpha. \tag{16}$$

An example of a 2-D pattern spectrum is shown in Fig. 2(b). Each dimension matches a specific attribute [22] with the axes normalized based on some mapping function. This, for the experiments that follow, is the simple area mapper from [22]. After initializing the n-dimensional pattern spectrum matrix, the binning function configured for each attribute separately, scales the component attribute based on the chosen mapping function and identifies the bin to which it contributes. The scaled contribution of each component, i.e. the term "area$(C_k^\alpha)\Delta\alpha$" from (16), is accumulated to the energy counter of each bin. The latter is gray level coded.

6 Applications

Pattern spectra have been used as feature vectors to describe image objects or regions in both classification and regression problems. An example highlighting



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

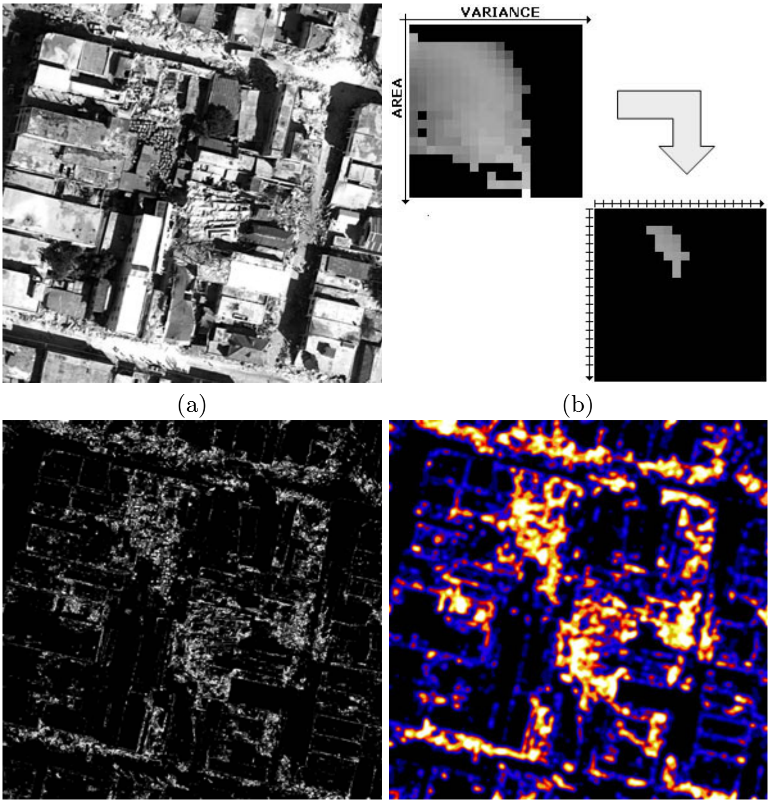
Fig. 3. Diatom ornament analysis: The original image in (a) with its pattern spectrum at the top left corner. The attributes along the x and y-axis are non-compactness and ω-range respectively. The image structures corresponding to bins (3,5), (4,5) and $(6,5) \cup (4,6) \cup (5,6)$ are highlighted in images (b), (c) and (d) respectively with the selected bins of the pattern spectra displayed at their top left corner.

the strength of this method in comparison to other descriptors, is in the classification of diatoms from microscopy imaging [3].

The work in [22] reports on the performance of 2-D pattern spectra computed based on the non-compactness and area attributes of connected components. The method outperforms all others from [3] but reaches an upper bound in classification success due to limitations in its descriptive power. The problem, that is addressed in [12], is essentially the inability of connected operators configured with regular adjacency-based connectivity, to deal effectively with component subregions that are more 'loosely' connected. Operators configured with partition-induced connectivity deal with this issue more effectively [12] and yield reacher spectra. This is at the cost of a higher computational demand due to the absence of nesting properties.

Following is a demonstration of a pattern spectrum computed based on the non-compactness and total range (ω) attributes of α-CCs. Image (a) in Fig.3 shows a sample from the ADIAC database [3] with its pattern spectrum at the top left corner. Images (b-d) show selected object features (diatom ornament and contour fragments) that correspond to some indicative bins of the α-connected pattern spectrum. We observe robust extraction of distinct features which suggests a better descriptive power for the proposed method.

The second application concerns rubble detection and follows the basic strategy from [10], only dealing with a selection of α-CCs instead of standard connected components. In brief, the method can be summarized in two steps; the



(a) (b)

(c) (d)

Fig. 4. Rubble detection based on selected image features from the α-connected pattern spectrum. The original image in (a), its pattern spectrum (left) and selected bins (right) in (b). The image features corresponding to the selected bins (c) and their color coded spatial aggregation in (d).

accumulation of the first plane of the opening and closing instance of a DMP vector field [13] configured with the area attribute [8] followed by spatial aggregation by a Gaussian low pass filter.

In this example, an α-connected pattern spectrum is computed based on the variance and area attributes. Rather than dealing with explicit attribute classes we chose a range of spectral bins. For the size attribute the bins range from the size corresponding to a quarter of the regular building size (estimated empirically) up to its full size. The range of the variance attribute is kept short since the rubble fragments are expected to be of poor texture. It is only the local background that is highly textured. Image (a) of Fig. 4 shows a sample of aerial, very high resolution imagery (approximately 15cm) of Port-au-Prince after the earthquake in Haiti in January 12^{th} 2010. The image is courtesy of Google 2010 and is available at the Google Crisis Response web-site at `http://www.google.com/relief/haitiearthquake/imagery.html`. Image

(b) shows the original α-connected pattern spectrum (top) and the selected bins (bottom). Image (c) shows the features corresponding to the selected bins from (b). Note that the color of each α-CC is the average of all 0-CCs it contains. The result of a low pass Gaussian filter with a square kernel size of 41×41 in thermal palette display is shown in image (d). The kernel size is set to match the estimated length of the average building size, i.e. approximately 6.5m.

7 Conclusions

Constraining dissimilarity-based connectivity relations is a way of controlling the leakage effect induced by connective criteria, discussed in Section 3. Constraints were used to control the evolution of single-linkage components along a dissimilarity range [19]. This approach however was shown to be limited to specific attribute thresholds and does not allow interactivity. In this paper we presented two hierarchically-ordered partition representation structures that map the sequence of progressive partitions along the dissimilarity range in the absence of constraints. In this approach, the type and number of constraints and the threshold value associated to each one separately, may be adjusted interactively. Moreover, through the hierarchical representation, we retain information on the evolution of the single-linkage independent on the threshold value. This was utilized in the the computation of α-connected pattern spectra and two application examples were given. We observed a high descriptive power of the proposed type of spectra, which remains to be further verified by large scale experimentation. Moreover, with both applications in the previous section using the slope as a dissimilarity measure, investigating the suitability and usage of others such as those presented in [4,20] is an open challenge. The hierarchical partition representation that was introduced suggests that a new tree-based representation algorithm can be developed that would allow the rapid processing of large datasets. This is currently under development [9], while in future work we aim at investigating the morphological-profile [13,8] equivalent feature vectors directly from α-hierarchies.

References

1. Braga-Neto, U.M., Goutsias, J.: Connectivity on complete lattices: New results. Computer Vision and Image Understanding 85(1), 22–53 (2002)
2. Breen, E.J., Jones, R.: Attribute openings, thinnings and granulometries. Computer Vision and Image Understanding 64(3), 377–389 (1996)
3. Buf, J.M.H.D., Bayer, M.M. (eds.): Automatic Diatom Identification. Series in Machine Perception and Artificial Intelligence. World Scientific Publishing Co., Singapore (2002)
4. Gueguen, L., Soille, P.: Frequent and dependent connectivities. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 120–131. Springer, Heidelberg (2011) (in press)
5. Maragos, P.: Pattern spectrum and multiscale shape representation. IEEE Trans. Pattern Analysis and Machine Intelligence 11(7), 701–715 (1989)

6. Meyer, F., Maragos, P.: Nonlinear scale-space representation with morphological levelings. Journal Visual Communication and Image Representation 11(3), 245–265 (2000)
7. Nagao, M., Matsuyama, T., Ikeda, Y.: Region extraction and shape analysis in aerial photographs. Computer Graphics and Image Processing 10(3), 195–203 (1979)
8. Ouzounis, G.K., Soille, P.: Differential area profiles. In: 20th Int. Conf. on Pattern Recognition, ICPR 2010, Istanbul, Turkey, pp. 4085–4088 (August 2010)
9. Ouzounis, G.K., Soille, P.: Attribute constrained connectivity and the alpha-tree representation. IEEE Trans. on Image Processing (submitted, 2011)
10. Ouzounis, G.K., Soille, P., Pesaresi, M.: Rubble detection from VHR aerial imagery data using differential morphological profiles. In: 34th Int. Symp. Remote Sensing of the Environment, Sydney, Australia (April 2011) (in press)
11. Ouzounis, G.K., Wilkinson, M.H.F.: Mask-based second-generation connectivity and attribute filters. IEEE Trans. Pattern Analysis and Machine Intelligence 29(6), 990–1004 (2007)
12. Ouzounis, G.K., Wilkinson, M.H.F.: Partition-induced connections and operators for pattern analysis. Pattern Recognition 43(10), 3193–3207 (2010)
13. Pesaresi, M., Benediktsson, J.A.: A new approach for the morphological segmentation of high-resolution satellite imagery. IEEE Trans. Geoscience and Remote Sensing 39(2), 309–320 (2001)
14. Ronse, C.: Set-theoretical algebraic approaches to connectivity in continuous or digital spaces. Journal of Mathematical Imaging and Vision 8, 41–58 (1998)
15. Ronse, C.: Partial partitions, partial connections and connective segmentation. Journal of Mathematical Imaging and Vision 32(2), 97–125 (2008)
16. Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Trans. on Image Processing 4(8), 1153–1160 (1995)
17. Serra, J. (ed.): Image Analysis and Mathematical Morphology. Theoretical Advances, vol. II. Academic Press, London (1988)
18. Serra, J.: A lattice approach to image segmentation. Journal of Mathematical Imaging and Vision 24, 83–130 (2006)
19. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Trans. Pattern Analysis and Machine Intelligence 30(7), 1132–1145 (2008)
20. Soille, P.: Preventing chaining through transitions while favouring it within homogeneous regions. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 96–107. Springer, Heidelberg (2011)
21. Soille, P.: On genuine connectivity relations based on logical predicates. In: Int. Conf. Image Analysis and Processing (ICIAP), pp. 487–492 (2007)
22. Urbach, E.R., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. IEEE Trans. Pattern Analysis and Machine Intelligence 29(2), 272–285 (2007)

Frequent and Dependent Connectivities

Lionel Gueguen and Pierre Soille

IPSC Global Security and Crisis Management Unit
Joint Research Centre, European Commission
Via E. Fermi, 2749, I-21027 Ispra (Va), Italy
name.surname@jrc.ec.europa.eu
http://isferea.jrc.ec.europa.eu/

Abstract. A dissimilarity measure between adjacent pixels of an image is usually determined by the intensity values of these pixels and therefore does not depend on statistics computed over the whole image domain. In this paper, new dissimilarity measures exploiting image statistics are proposed. This is achieved by introducing the notion of dissimilarity function defined for every possible pair of intensity values. Necessary conditions for generating a valid dissimilarity function are provided and a series of functions integrating image statistics are presented. For example, the joint probability of adjacent pixel values leads to the notion of frequent connectivity while the notion of dependent connectivity relies on the local mutual information. The usefulness of the proposed approach is demonstrated by a series of experiments on satellite image data.

Keywords: Connective segmentation, Single-Linkage Clustering, Dissimilarity Measures, Statistics, Mutual information.

1 Introduction

A connectivity relation between elements of an arbitrary set is an equivalence relation producing a unique partition of this set. The partition cells are equivalence classes that correspond to connected regions of maximal extent called connected components. For example, the single linkage clustering [1] corresponds to a connectivity relation where two objects are said to be connected if a sequence of objects starting by one of these two objects and ending by the other one can be found while ensuring that the dissimilarity measure between two successive objects of the sequence does not exceed a pre-defined dissimilarity threshold value. Single-linkage applies directly to digital images [2]. In this case, the objects under study are the pixels and it is assumed that non-adjacent pixels are infinitely dissimilar. Typically, two pixels of a grey tone image are connected if there exists a sequence of adjacent pixels (i.e., path) linking these pixels and such that the grey level absolute difference between successive pixels of the path (i.e., weights of the edges of the path) does not exceed a given threshold value [2,3]. The resulting connected components are sometimes called quasi-flat [4], lambda-flat zones [5] or, more recently, α-connected components [6]. Note that the absolute

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 120–131, 2011.

difference measure corresponds to the most natural dissimilarity measure between adjacent pixels. In general, the use of a dissimilarity measure d leads to d_α-connected components, in which α is a given dissimilarity threshold value.

In this paper, new dissimilarity measures between pairs of adjacent pixels are defined and exploited to define new connectivity relations. These dissimilarity measures are derived from image statistics, making the derived image partitions controlled by these statistics. Several techniques for defining suitable decision functions are explored. Frequent connectivity is based on image statistics related to the co-occurence of intensity values of adjacent pixels. Dependent connectivity goes one step further by measuring the statistical dependence between adjacent pixels using the concept of local mutual information. We demonstrate on a series of experiments the usefulness of these approaches.

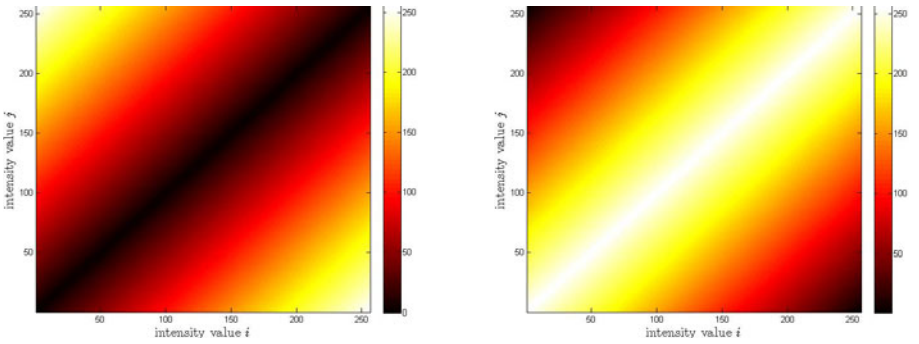
2 Preliminaries

Let I be a monochannel digital image and $\mathcal{G} = (V, E)$ an undirected graph with vertex set V matching the image pixels and edge set E consisting of unordered pairs of vertices indicating the adjacency relations between the image pixels (for example, 4 or 8-connected graphs for 2D images on the square lattice, 6 or 18-connected graphs for 3D images on the cubic lattice, and Delaunay graph for images sampled at irregularly spaced points). A path P between two pixels p and q in $\mathcal{G}$ is a sequence of $n > 1$ pixels $\langle p = p_1, \dots, p_n = q \rangle$ such that any two successive pixels of the sequence are adjacent: $\{p_i, p_{i+1}\} \in E$ for $1 \leq i < n$. The image I assigns to each element of V (that is, the pixels), a non-negative integer defining its intensity value (grey level).

The partition of a graph $\mathcal{G} = (V, E)$ into connected components relies on a function associating a weight w to its edges. This function can be viewed as a measure of the degree of dissimilarity between adjacent vertices. In this paper, we consider dissimilarity measures obtained through a dissimilarity function defined for every possible pair of intensity values.

Let $\mathcal{I}$ be the space of image intensity values. A dissimilarity function indexed by the image intensity values is defined as any function d from $\mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}^+$ such that $d(I(p_1), I(p_2)) = d(I(p_2), I(p_1))$. This latter property is imposed by the symmetry property of dissimilarity measures. The most natural dissimilarity function is obtained by considering the absolute difference of the input intensity values: $d^A(I(p_1), I(p_2)) = |I(p_1) - I(p_2)|$. This function is illustrated in Fig. 1(a) in the case of a byte image. Any other symmetric function may be considered such as the complement of the latter function, i.e., $d^C(I(p_1), I(p_2)) = \max_{p \in V}\{I(p)\} - |I(p_1) - I(p_2)|$, as displayed in Fig. 1(b).

Given a dissimilarity function d, the weight of an edge of the adjacency graph underlying an image I is denoted by w_d: $w_d(\{p, q\}) = d(I(p), I(q))$. Given a dissimilarity threshold α, two distinct pixels p and q of an image I are d_α-connected if there exists a path going from p to q such that the weight w_d between any two successive pixels of this path does not exceed the value of the dissimilarity threshold level α. In addition, to ensure the reflexivity property of an equivalence



(a) $d^A(i,j)$

(b) $d^C(i,j)$

Fig. 1. Two examples of dissimilarity functions indexed by the intensity values i and j of two arbitrary adjacent pixels. a) The absolute difference of grey levels $d^A(i,j) = |i-j|$. b) The complement of the absolute difference function $d^C(i,j) = 255 - |i-j|$, where 255 is the maximal grey level.

relation, a pixel is always said to be d_α-connected to itself. The d_α-connectivity being an equivalence relation, it induces a unique partition of the image support into maximal connected regions called d_α-connected components. Accordingly, the d_α-connected component of a pixel p is defined as the set of image pixels that are d_α-connected to this pixel. We denote this connected component by d_α-CC(p):

$$\begin{aligned} d_\alpha\text{-CC}(p) = \{p\} \cup \{q \ \mid \exists\, \mathcal{P} = \langle p = p_1, \ldots, p_n = q\rangle, \\ d(I(p_i), I(p_{i+1})) \leq \alpha, \forall\, 1 \leq i < n\}. \end{aligned} \tag{1}$$

When using the dissimilarity function based on the absolute difference of the input pixel values d^A, as originally proposed in [2], the resulting d_a^A-connected components are sometimes called quasi-flat zones [4], lambda-flat zones [5], or α-connected components [6].

A fundamental property of the d_α-connected components of a pixel is that they form an ordered sequence (hierarchy) when increasing the dissimilarity threshold value α:

$$d_{\alpha_1}\text{-CC}(p) \subseteq d_{\alpha_2}\text{-CC}(p), \ \forall \alpha_1 \leq \alpha_2. \tag{2}$$

This hierarchy is at the root of the greedy algorithm by Kruskal [7] for solving the minimum spanning tree problem and at the very basis of the dendrogram representation of the single linkage clustering [8].

Rather than using the notion of equivalence relation and path-based connectivity, Serra [9,10] proposes to study connectivity by introducing the notion of connection and connective criterion. A connection $\mathcal{C}$ on an arbitrary non-empty space Ω is any family of the power set of Ω such that (i) $\emptyset \in \mathcal{C}$, (ii) for all $x \in \Omega$, $\{x\} \in \mathcal{C}$, and (iii) any family of elements $C_i \in \mathcal{C}$ such that $\cap_i C_i \neq \emptyset$

implies that $\cup_i C_i \in \mathcal{C}$. Note that properties (i) and (ii) are analogous, respectively, to the reflexivity and transitivity properties of an equivalence relation. A Boolean criterion σ on the power set of Ω is then said to be connective if the elements of this power set for which the criterion is satisfied generate a connection. Accordingly, the Boolean connective criterion underlying d_α-connectivity is defined as follows: $\sigma_{d_\alpha}(C_i) =$ true if and only if for all $p, q \in C_i$ there exists a path $\langle p = p_1, \ldots, p_n = q\rangle$ such that $d(I(p_i), I(p_{i+1})) \leq \alpha$ for all $i \in \{1, \ldots, n-1\}$. By definition, $\sigma_{d_\alpha}(C_i)$ is also true for every C_i reduced to a single pixel. Segmentations expressed in terms of connective criteria are referred to as connective segmentations [10].

3 Adaptive Connectivity

Generally, connectivity relations are independent of statistics computed over the whole image definition domain. They are rather defined to model some class of image, such as k-Lipschitz functions [10]. In this section, we present new connectivity relations based on the definition of new dissimilarity functions exploiting image statistics. Such connectivity relations induce partitions that are adaptive to the image content.

3.1 Frequent Connectivity

Inspired by the analysis of transition frequencies between adjacent regions developed in [11], we propose to use Grey Level Cooccurrence Matrix (GLCM) [12] in order to derive joint probability of adjacent pixels. The GLCM M of an image I is obtained by:

$$M(i,j) = \sum_{\{p,q\}\in E} \max\{\mathbb{1}(I(p) = i, I(q) = j), \mathbb{1}(I(q) = i, I(p) = j)\}, \qquad (3)$$

where E is the set of edges defining the graph of image grid, and $\mathbb{1}$ is an indicator function. By construction, the GLCM M is positive and symmetric so that it defines a valid dissimilarity function. The GLCM M can be transformed into a symmetric joint probability distribution defined over $\mathcal{I}^2$ by:

$$p(i,j) = \frac{M(i,j)}{|E|}, \qquad (4)$$

where $|E|$ is the number of edges which are contained in the graph of the image grid. The joint probability distribution highlights frequent co-occurring grey values pairs and has been widely used in literature for characterizing textures [12]. The neighbor grey values appearing frequently indicate a high repetition of this pattern in the image. We propose to define a dissimilarity where frequent co-occurring intensities are similar. The frequent dissimilarity d^F is defined by:

$$d^F(i,j) = 1 - p(i,j). \qquad (5)$$

Having a threshold f, all the adjacent pixels having their frequent dissimilarity greater than f are considered connected. The induced connected components forming a segmentation of the image support are denoted by d^F_f-CC.

An example of the dissimilarity function d^F is given in Fig. 2(b). One can observe, that this function behaves like grey level absolute difference d^A (see Fig. 1(a)), as close grey values are likely to happen. However, the frequent dissimilarity tends to connect first the intermediate grey values, then the extremal ones.

Due to the increasing property (2), the d^F_f-CC form an ordered sequence when increasing the threshold value f:

$$d^F_{f_1}\text{-CC}(p) \subseteq d^F_{f_2}\text{-CC}(p),\ \forall f_1 \leq f_2. \tag{6}$$

As mentioned previously, this property enables the definition of hierarchical segmentation by increasing the threshold value.

3.2 Dependent Connectivity

In this section, one step ahead frequent connectivity is taken by analyzing the statistics of adjacent grey levels. More than frequency, independence analysis looks for statistical links between variables. Thus, the grey level pairs having a high statistical link (dependence) are considered to be connected. In order to analyze adjacent pixels dependence, we use the local mutual information function as defined in [13,14] and expressed by:

$$\mathrm{mi}(i,j) = \log\left(\frac{p(i,j)}{p(i)p(j)}\right), \tag{7}$$

where $p(i) = \sum_{j\in\mathcal{I}} p(i,j)$ and $p(j) = \sum_{i\in\mathcal{I}} p(i,j)$, $p(i,j)$ being the joint distribution expressed in (4). The function $\mathrm{mi}(i,j)$ locally measures the dependence of adjacent grey levels i and j, by comparing the estimated joint distribution $p(i,j)$ to the corresponding independent joint distribution $p(i)p(j)$. Thus, a high local mutual information highlights a strong dependence between adjacent intensity values. The dependent dissimilarity metric d^D is then expressed by:

$$d^D(i,j) = \max_{(k,l)\in\mathcal{I}\times\mathcal{I}} \{\mathrm{mi}(k,l)\} - \mathrm{mi}(i,j). \tag{8}$$

Such a definition ensures that d^D is positive and an example of the dependent dissimilarity function is given in Fig. 2(c). One can observe that the dependent dissimilarity is also close to the absolute difference of grey values, while enabling more connectivity for extremal image values.

Given a threshold m, the induced dependent connected components are denoted by d^D_m-CC. They also form an ordered sequence when increasing the value m because of the property (2).

4 Results and Discussion

In this section, visual and quantitative analysis of the proposed dissimilarity based segmentation are conducted. In the analysis, the given dissimilarity functions and resulting connected components are compared to the d_a^A-connected components.

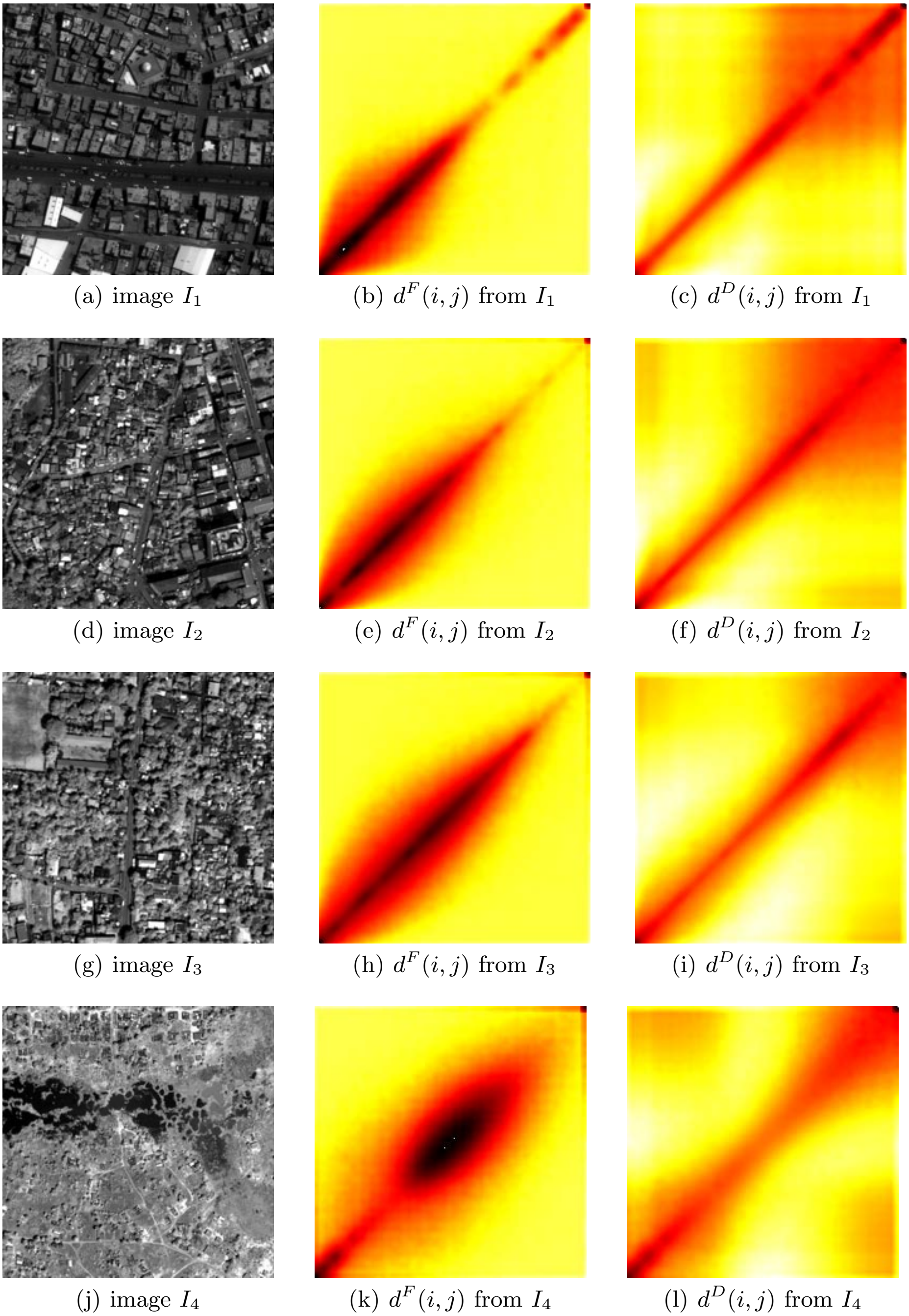
4.1 Visual Analysis of the d^F and d^D Dissimilarity Functions

The frequent and dependent dissimilarity functions derived from four different images are displayed in Fig. 2. The images are subsets of very high resolution remote sensing images. The dissimilarities depend on and adapt to the image statistics, which vary significantly. However common effects are visible for each image. While the frequent connectivity tends to assimilate the intermediate grey values, the dependent connectivity tends to gather the extremal intensity values. This last connectivity limits the connectivity between intermediate intensity values which are in majority responsible for chaining through transitions (leakage effect).

4.2 Visual Analysis of the d^A, d^F and d^D Based Segmentations

To assess the impact of the chosen dissimilarities, the connected components of the image displayed in Fig. 3 are computed for the absolute difference, frequent and dependent dissimilarity functions. The image was acquired by the very high resolution optical sensor QuickBird, and has a resolution of 60cm. Suitable thresholds have been selected in such a way that all considered functions have a similar number of connections in the image (i.e., similar sum of probabilities over the true decision region). Comparing both dissimilarity functions with the absolute difference (Fig. 4(a)), we observe that they are similar in the sense that connections resides along the small differences between grey values. However, the proposed dissimilarity functions vary the impact of the grey value differences along the equal grey values line. In the frequent connectivity case (Fig. 4(b)), the connectivity of black or white pixels is limited, while in the dependent connectivity case (Fig. 4(c)), the connectivity of intermediate grey values is limited. In this last case, for a sufficiently low threshold, connection between white and black values appears. This phenomenon is due to the image nature, where lot of strong edges exist. The frequent connectivity produces fewer connected components (Fig. 4(e)), where one big connected component covers the major image support. In between, d_a^A-CC and d_m^D-CC produce segmentations (Figs. 4(d)-4(f)) with similar numbers of connected components. Still the d_m^D-CC based partition produces more connected components than the d_a^A-CC based partition. Further examples are provided in Fig. 5, where four very high resolution satellite images are considered. The same comments made above hold for the four images containing varying information content.

From this visual analysis, one can argue that dependent connectivity reduces chaining through ramps which produces coarse partitions for the following reasons. The frequent connectivity suffers from chaining through ramps, since one


(a) image I_1

(b) $d^F(i,j)$ from I_1

(c) $d^D(i,j)$ from I_1

(d) image I_2

(e) $d^F(i,j)$ from I_2

(f) $d^D(i,j)$ from I_2

(g) image I_3

(h) $d^F(i,j)$ from I_3

(i) $d^D(i,j)$ from I_3

(j) image I_4

(k) $d^F(i,j)$ from I_4

(l) $d^D(i,j)$ from I_4

Fig. 2. The frequent d^F and dependent d^D dissimilarity functions corresponding to the images displayed in (a), (d), (g), and (j)

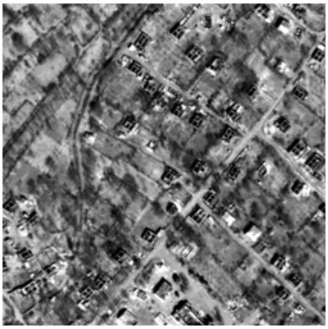
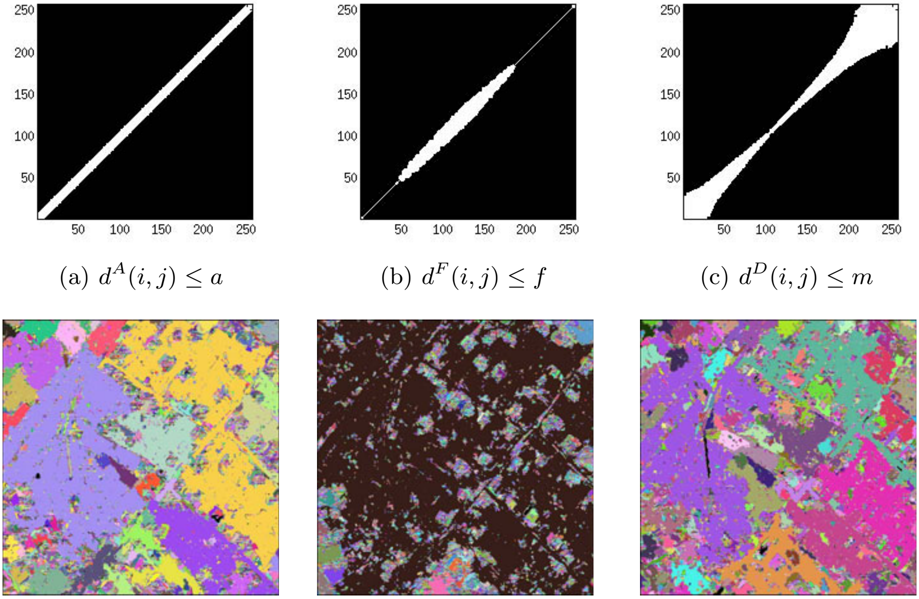


Fig. 3. Very High Resolution optical panchromatic image of size 512×512 pixels acquired by QuickBird sensor. Credit DigitalGlobe 2009.



(a) $d^A(i,j) \leq a$

(b) $d^F(i,j) \leq f$

(c) $d^D(i,j) \leq m$

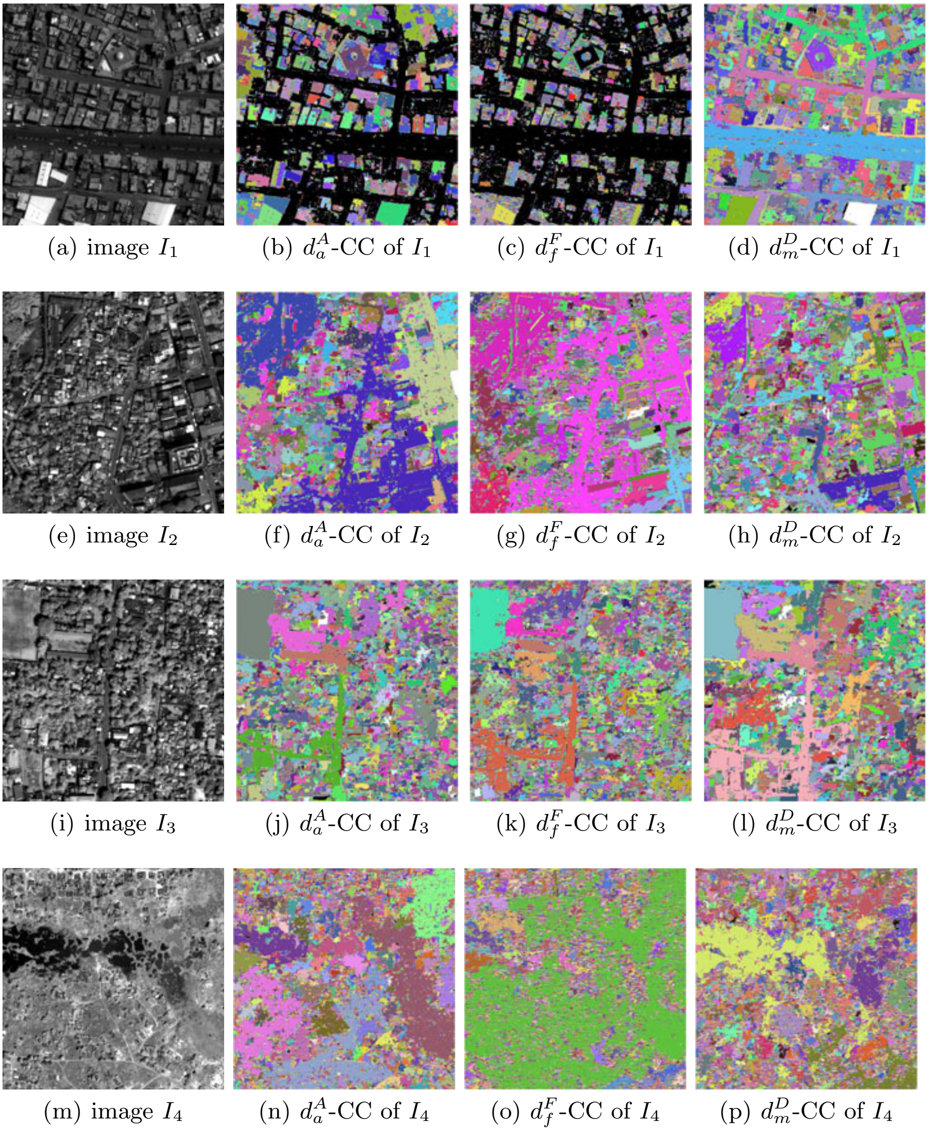
(d) d^A_a-CC

(e) d^F_f-CC

(f) d^D_m-CC

Fig. 4. The thresholded dissimilarity functions are displayed in (a), (b), (c) and they are derived from the image represented in Fig. 3. The corresponding connected components are displayed in (d), (e), (f), respectively.

big component is produced. Analyzing where connexions are favoured, it appears that connections between intermediate grey values facilitate this effect. On the contrary, the dependent connectivity is more robust to chaining through ramps since it limits the connectivity between intermediate intensity values. From this prospective, it seems that the dependent connectivity provides an interesting alternative to the d^A-connectivity.



(a) image I_1 (b) d_a^A-CC of I_1 (c) d_f^F-CC of I_1 (d) d_m^D-CC of I_1

(e) image I_2 (f) d_a^A-CC of I_2 (g) d_f^F-CC of I_2 (h) d_m^D-CC of I_2

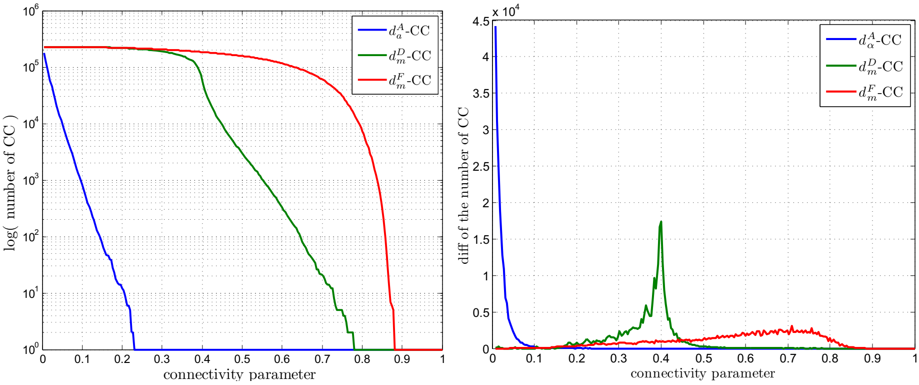
(i) image I_3 (j) d_a^A-CC of I_3 (k) d_f^F-CC of I_3 (l) d_m^D-CC of I_3

(m) image I_4 (n) d_a^A-CC of I_4 (o) d_f^F-CC of I_4 (p) d_m^D-CC of I_4

Fig. 5. Various satellite images with their corresponding segmentations based on d^A, d^F, and d^D dissimilarity functions.

4.3 Quantitative Analysis the d^A, d^F and d^D Based Segmentations

In this section, a quantitative analysis of images components is made by calculating the impact of the connectivity parameters on the number of connected components. Increasing the connectivity parameter induces hierarchical segmentations as described by ([2]). This hierarchical property was proven to be relevant



(a) number of CC

(b) differences of the number CC

Fig. 6. The evolution of the number of CC and their successive differences highlighting the diversity of the hierarchical partitions

to define adapted connected components selection, through the definition of additional constraints [15]. Representing the hierarchy as a tree, where each node is a connected component obtained for some connectivity parameter, considering only the connectivities proposed in this paper consists in performing horizontal cuts in the tree. On the contrary, enabling additional constraints enables to perform adapted cuts in the tree. Therefore, having a high diversity of connected components in the hierarchical tree is an advantage in order to obtain a desired partition or to perform an optimization [16]. Indeed, the space of possible partitions becomes larger as the diversity of connected components increases. We propose to assess the diversity of the tree by evaluating the differences of the number of connected components from one threshold to the successive one. Then, a type of connectivity which produces large differences by slightly changing the connectivity parameter (a, f or m) is not diverse since it misses many intermediate states. On the contrary, a type of connectivity which produces small differences is interesting because it goes through intermediate states, which indicates the existence of more connected components in the hierarchy. In Fig. 6(a), the evolution of the number of connected components with the connectivity parameter are displayed for the sets d_a^A-CC, d_f^F-CC, and d_m^D-CC, where the connectivity parameters a, f and m are normalized between 0 and 1. In Fig. 6(b), the differences of numbers of CC are displayed. d^A-connectivity produces the biggest differences, highlighting a lower diversity than other connectivities. In the case of d^F-connectivity, chaining through transitions (leakage) was observed previously, such that the partitions are composed of big components and many small ones composed of single pixels. The slow variation of the number of d_f^F-CC is due to the connexion of these isolated pixels. Thus, d^D-connectivity constitutes a good compromise between the diversity of connected components and the leakage effect.

5 Concluding Remarks

We have presented new connectivity definitions in order to produce hierarchical partitions of images. The proposed connectivity definitions are derived from the statistics of the image, and are based on the analysis of adjacent pixels coocurrences. One type of connectivity considers the frequency of adjacent grey values, while the second one makes use of the dependence of adjacent grey values. A visual assessment of the various connectivities is performed on satellite images, highlighting that d_m^D-CC limits the single linkage chain effect. Then, a quantitative analysis of the number of connected components shows the higher diversity induced by the dependent connectivity, which makes it to be an interesting alternative to d_a^A-CC. Finally, we plan to define new constrained connectivity based on the d_m^D-CC, trying to exploit the bigger diversity of the produced connected components, and to compare the effectiveness of the method for mitigating chaining through ramps with the methods described in [17].

References

1. Sneath, P.: The application of computers in taxonomy. Journal of General Microbiology 17, 201–226 (1957)
2. Nagao, M., Matsuyama, T., Ikeda, Y.: Region extraction and shape analysis in aerial photographs. Computer Graphics and Image Processing 10(3), 195–223 (1979)
3. Baraldi, A., Parmiggiani, F.: Single linkage region growing algorithms based on the vector degree of match. IEEE Transactions on Geoscience and Remote Sensing 34(1), 137–148 (1996)
4. Meyer, F., Maragos, P.: Morphological scale-space representation with levelings. In: Nielsen, M., Johansen, P., Fogh Olsen, O., Weickert, J. (eds.) Scale-Space 1999. LNCS, vol. 1682, pp. 187–198. Springer, Heidelberg (1999)
5. Zanoguera, F., Meyer, F.: On the implementation of non-separable vector levelings. In: Talbot, H., Beare, R. (eds.) Proc. of VIth ISMM, Sydney, CSIRO, pp. 369–377 (2002)
6. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(7), 1132–1145 (2008)
7. Kruskal, J.: On the shortest spanning subtree of a graph and the traveling salesman problem. Proc. Am. Math. Soc. 7(1), 48–50 (1956)
8. Gower, J., Ross, G.: Minimum spanning trees and single linkage cluster analysis. Applied statistics 18(1), 54–64 (1969)
9. Serra, J.: Mathematical morphology for Boolean lattices. In: Serra, J. (ed.) Image Analysis and Mathematical Morphology. Theoretical Advances, vol. 2, pp. 37–58. Academic Press, London (1988)
10. Serra, J.: A lattice approach to image segmentation. Journal of Mathematical Imaging and Vision 24, 83–130 (2006)
11. Akçay, G., Aksoy, S., Soille, P.: Hierarchical segmentation of complex structures. In: Proc. of 20th Int. Conf. on Pattern Recognition, Istanbul, pp. 1120–1123. IEEE, Los Alamitos (2010)

12. Haralick, R., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics 3(6), 610–621 (1973)
13. Cover, T., Thomas, J.: Elements of Information Theory, 2nd edn. Wiley & Sons, Chichester (2006)
14. Gueguen, L., Datcu, M.: Mixed information measure: Application to change detection in earth observation. In: MultiTemp 2009: The Fifth International Workshop on the Analysis of Multi-temporal Remote Sensing Images, Connecticut, USA (June 2009)
15. Soille, P.: On genuine connectivity relations based on logical predicates. In: Proc. of 14th Int. Conf. on Image Analysis and Processing, Modena, Italy, pp. 487–492. IEEE Computer Society Press, Los Alamitos (2007)
16. Panjwani, D.K., Healey, G.: Markov random field models for unsupervised segmentation of textured color images. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(10), 939–954 (1995)
17. Soille, P.: Preventing chaining through transitions while favouring it within homogeneous regions. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 96–107. Springer, Heidelberg (2011)

Stochastic Multiscale Segmentation Constrained by Image Content

Luc Gillibert and Dominique Jeulin

Centre de Morphologie Mathematique
Ecole des Mines de Paris
35, rue Saint Honoré
77305 Fontainebleau, France

Abstract. We introduce a noise-tolerant segmentation algorithm efficient on 3D multiscale granular materials. The approach uses a graph-based version of the stochastic watershed and relies on the morphological granulometry of the image to achieve a content-driven unsupervised segmentation. We present results on both a virtual material and a real X-ray microtomographic image of solid propellant.

Keywords: Granular media, granulometry, multiscale 3D segmentation, stochastic watershed, constrained segmentation.

1 Introduction

The stochastic watershed segmentation was first introduced by Angulo and Jeulin in [1]. The approach is based on using a large number of realizations of random markers to build a probability density function (pdf) of contours, starting from a standard watershed algorithm producing oversegmentations.

The stochastic watershed was proved to be efficient for unsupervised segmentation [10][6]. The two parameters used for its construction are k, the number of random markers used in each realization, and R, the number of realizations. From the law of large numbers, the pdf converges when increasing R. The parameter k needs to be proportional to the number of desired regions in the segmented image. Therefore, in the case of granular materials, k needs to be proportional to the number of grains contained in the image, that can be automatically estimated. In [6], the authors use the covariance for this estimation, deduced from the average radius of the grains, and a Boolean model assumption [8].

For multiscale images, the covariance is not so efficient. In this paper, we introduce a new approach for stochastic segmentation which relies on the full granulometry of the image. Using morphological openings, this granulometry can be automatically computed from the image and is used as a constraint during iterations of segmentation steps.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 132–142, 2011.

2 Stochastic Segmentation

2.1 Stochastic Watershed

The first method introduced for computing the stochastic watershed is based on a large number of realizations of random markers to estimate a pdf of contours, or of surface boundaries in 3D. The random markers are generated with an uniform distribution corresponding to a constant intensity. In the case of granular materials, a constant background marker, extracted by an automatic thresholding, is added to each set of random markers. For each set of markers, a constrained watershed is computed. Then, the Parzen window is used to estimate the pdf of contours.

For a good estimation of the stochastic watershed, 100 to 200 realizations are required [1]. However, using λ-flat zones, a stochastic watershed segmentation can be achieved with only 50 realizations [6]. This number is low, but the computation of 50 watersheds is very time-consuming, especially on large 3D data sets.

From the pdf, it is possible to obtain the segmentation. The first approach uses this pdf as a gradient for a new watershed [1]. A more efficient approach uses λ-flat zones to overcome the fact that the estimated pdf is not constant over each branch of contour [6]. Illustration of the pdf of contours is on Fig. 1 (A). The resulting segmentation is illustrated on Fig. 2 (A).

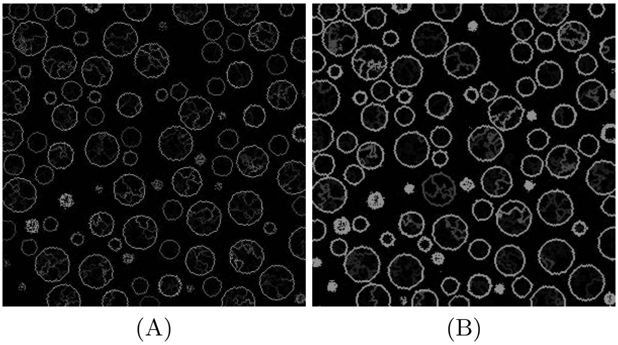


Fig. 1. (A) Pdf of contours on a simulated $3D$ material R_1, estimated with 50 realizations of a Poisson point process (slice). (B) Pdf of contours computed with a graph-based approach (slice).

2.2 Graph-Based Stochastic Watershed

Computing a large number of watersheds from simulations provides good results but is a slow process, mainly in $3D$. A more efficient solution for computing stochastic watersheds is to use a graph-based approach. Probability of boundaries is directly computed with a good approximation without the use of any realisation [9,7].

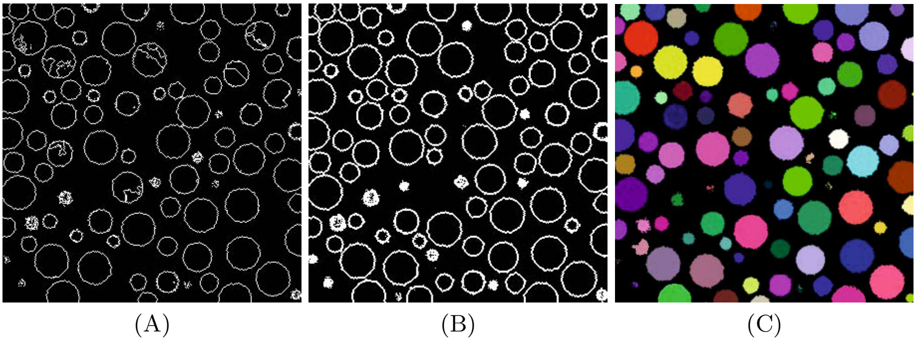


Fig. 2. (A) Stochastic watershed segmentation (slice). The λ-flat zones are used. A few grains are over-segmented. Small grains are missing (B) Graph-based stochastic watershed segmentation with a threshold of 0.5 on the probability (slice). A few over-segmented grains, as well as small grains are missing. (C) Final granulometry-driven multiscale segmentation result for the random material R_1 (slice of a $3D$ image).

A first watershed is computed from the local minima of the gradient, as in standard segmentation, but restricted to the complementary set of the background extracted by an automatic thresholding. For this purpose, the background is used as a marker. In the present case, a very strong oversegmentation is obtained as a result of the presence of noise. From this watershed, an adjacency graph is constructed. Vertices of the graph are associated to each basin of the watershed, connecting edges between adjacent regions. A vertex is associated to the background too. Values are given to the vertices corresponding to the volumes of the regions (or to the areas, in 2D). Each edge is labelled with the minimum of the gradient function on the boundary between the corresponding regions. From this valued graph, a minimum spanning tree is extracted [7]. Then the regions in the minimum spanning tree are merged, starting with the edge of lowest value. After each merging, the probability p of the boundary between the corresponding regions is estimated using the following equation [9]:

$$p = 1 - (1 - V_1/V)^k - (1 - V_2/V)^k + (1 - (V_1 + V_2)/V))^k \qquad (1)$$

This is the probability of obtaining at least one random marker in each of the two regions, knowing the volumes V_1 and V_2 of the two regions, the total volume V of the image and the number of markers k. It is seen from equation 1 that the probability p increases with the volume of the grain, that may cause a bias towards largest grains if there is a wide distribution of sizes in the image.

After the merging of all the nodes in the original minimum spanning tree, the probability of all the edges of the tree is known. The result is projected from the tree on the graph and from the graph on the image.

This approach provides uniform probability on each part of boundary between two regions, as illustrated on Fig. 1 (B). Therefore, the λ-flat zones are useless and a simple threshold can be used for the segmentation Fig. 2 (B).

3 Multiscale Image Segmentation

The multiscale image segmentation process is based on a simple idea: estimate the full granulometry of the image, using morphological openings, then use multiple stochastic watersheds with different numbers of markers for each size, and finally combine them to get a segmentation that is pertinent for each size of grains. For achieving this goal, a hierarchy on the boundaries of the stochastic watershed is required. Many hierarchical segmentation algorithms have been studied, as the waterfalls [3] or the P algorithm [4]. Here we introduce an approach similar to the one used for the computation of the waterfall: the merging of the watersheds basins using a minimum spanning tree [5].

3.1 Hierarchy on Boundaries

Each boundary is an edge in the adjacency graph. For a non-multiscale graph-based stochastic watershed, a fixed threshold is efficient. So, a first approach for constructing a hierarchy on the edges of the adjacency graph is to use a threshold. All the edges with a probability less than a given value t are removed from the graph. With $t \in [0, 1]$, a complete hierarchy is obtained [9].

This approach is fast and easy to compute, but, for a large value of t the remaining edges are not exploitable as boundaries for a segmentation. The removal of edges implies the removal of boundaries and therefore the merging of regions in the images. This process changes the probability of the boundaries of the remaining regions if we iterate the segmentation. This phenomenon is ignored with a fixed threshold. As seen on Fig. 2 (B), some over-segmented grains will disappear by a simple trheshold.

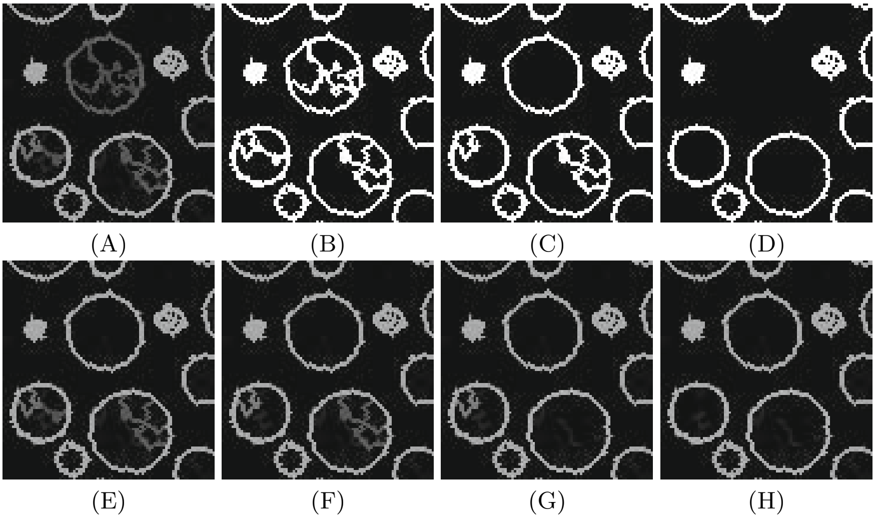
Working with a graph, it is possible to update the probability of the boundaries of the remaining regions at each edge deletion with the following iterative algorithm [9]:

- While the MST has at least 2 vertices.
  - In the MST, the edge e with the lowest probability is chosen.
  - The edge e is deleted and the adjacent vertices are merged.
  - The probability of the edges incident to the merged vertex is updated, using Eq. 1 and the new volume of the vertex.

This leads to better hierarchy on boundaries. As seen on Fig. 3, with the threshold approach some over-segmented grains disappear easily, while this problem is solved with the merging algorithm.

3.2 Granulometry-Driven Multiscale Approach

In order to get a pertinent segmentation starting from a highly over-segmented image, we can introduce constraints in the merging process. In the present case, the first step of the approach is to estimate the full granulometry of the image,


(A) (B) (C) (D)

(E) (F) (G) (H)

Fig. 3. (A) Probability density function of contours on a $3D$ material R_1, computed with a graph-based approach (detail of a slice). (B), (C), (D): hierarchy on boundaries using increasing thresholds. (A), (E), (F), (G), (H): hierarchy on boundaries using the merging algorithm, step 0 is (A).

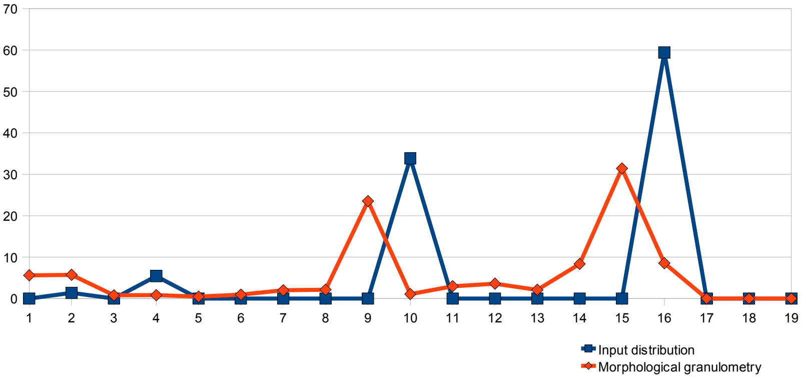
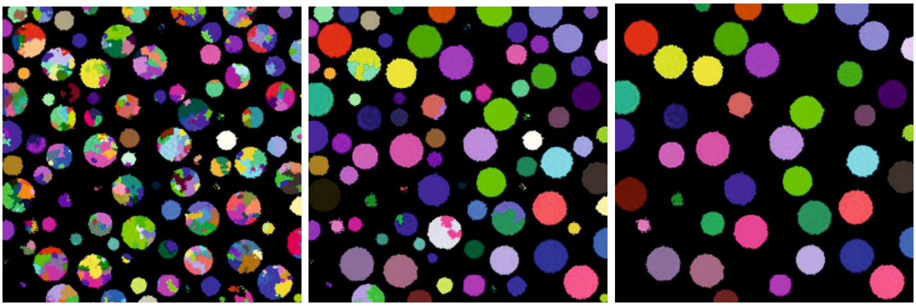


Fig. 4. Input granulometry of the random material R_1 and granulometry measured on the thresholded image from morphological openings. On x axis, the radius, and on y axis the volume fraction of grains with this radius (in percent).

using morphological openings. For fast computations, we use a rhombicuboctahedron as structuring element and we work on the binarized (thresholded) image. Gaussian noise, border effects and the shape of the structuring element leads to a few errors on the granulometry, as seen on Fig. 4, but they will not induce errors in the segmentation process.



(A) (B) (C)

Fig. 5. The segmentation of the random material R_1 (slice). Results for the class 1 on steps 1 (A), 20 (B) and 40 (C).

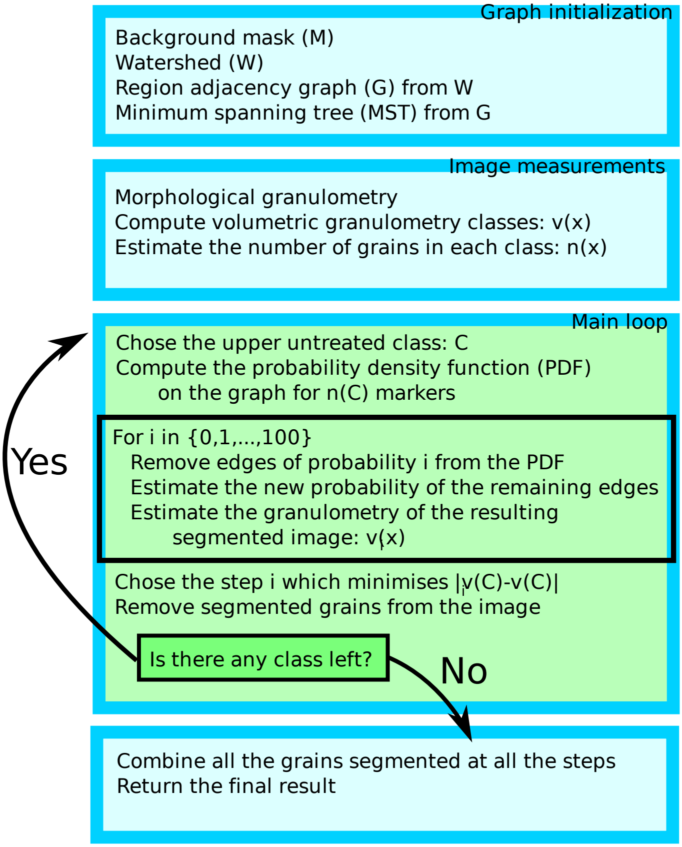

Graph initialization
Background mask (M)
Watershed (W)
Region adjacency graph (G) from W
Minimum spanning tree (MST) from G
Image measurements
Morphological granulometry
Compute volumetric granulometry classes: v(x)
Estimate the number of grains in each class: n(x)
Main loop
Chose the upper untreated class: C
Compute the probability density function (PDF)
on the graph for n(C) markers
For i in {0,1,...,100}
Remove edges of probability i from the PDF
Estimate the new probability of the remaining edges
Estimate the granulometry of the resulting
segmented image: $v_i(x)$
Chose the step i which minimises $|v_i(C)-v(C)|$
Remove segmented grains from the image
Is there any class left?
Yes
No
Combine all the grains segmented at all the steps
Return the final result

Fig. 6. The main steps of the multiscale image segmentation process

From the granulometry, a small number of classes are chosen. A good approach for this choice is maximising the interclass variance. For our sample material R_1, we uses 3 classes: radius [13,18], radius [6,13] and radius [1,5]. The total volume of the grains in each class will be denoted $v(x)$. The number of grains in each class is deduced from $v(x)$ and is denoted $n(x)$. It is used to generate the corresponding number of markers in the calculation of the probability.

Then, the standard watershed is computed from the local minima of the gradient. From this watershed, the adjacency graph is constructed and a minimum spanning tree is extracted.

The first class is chosen. The stochastic watershed is computed with a number of markers equals to $n(1)$. In our $3D$ example (material R_1), 417 markers are needed. Based on this stochastic watershed, a first hierarchy on boundaries is computed with the merging algorithm. For each step i of the hierarchy, the granulometry of the corresponding segmentation is computed $(v_i(1))$.

In the full hierarchy, there is a size step i^* which minimises the difference $|v_i(1) - v(1)|$. In our example, $i^* = 40$ (Fig. 11 and Fig. 5). This step is used for the segmentation of the grains in the first class. All the segmented grains are removed from the image and added to the background mask. The minimum spanning tree is updated and the next class is chosen.

The same process is applied to all the classes. When no more class are left, all the segmentations are combined together. This provides the final result illustrated on Fig. 2 (C) and Fig. 12. The main lines of the multiscale image segmentation process are summarized on Fig. 6.

4 Validation

4.1 Random Model

For the validation, we use a simple random model. We generate 7560 random discrete spheres. The distribution of the radius of the spheres is fixed and is given

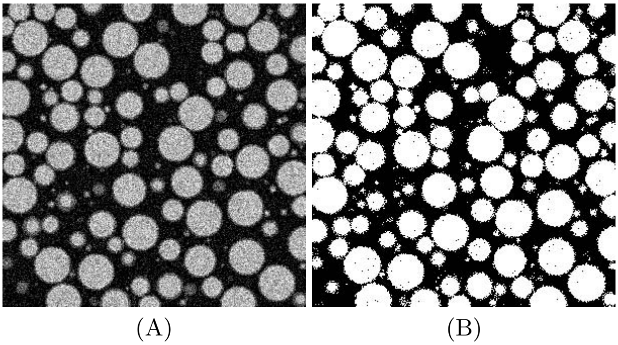


Fig. 7. (A) Random material R_1 with a strong Gaussian noise and a wide grain size distribution (slice). (B) The binary mask for material R_1 (slice), the threshold is calculated via the maximization of the interclass variance.

on Fig. 4. The centres of the spheres are uniformly distributed in a cubic volume of size 300 × 300 × 300 with a rejection sampling algorithm (acceptance-rejection method). Two spheres cannot overlap more than a given threshold (fixed to 2 voxels). Spheres have non-uniform grey values, and show a visible boundary generating information in the gradient image. The volume fraction of the grains on the sample is 0.432. A strong Gaussian noise is finally added. A slice of the simulated material and of the corresponding binary mask is illustrated on Fig. 7.

4.2 Microtomographic Images

Our second sample is an X-ray microtomographic image provided by the CEA Gramat. The image was generated at with a Skyscan 1172 high-resolution

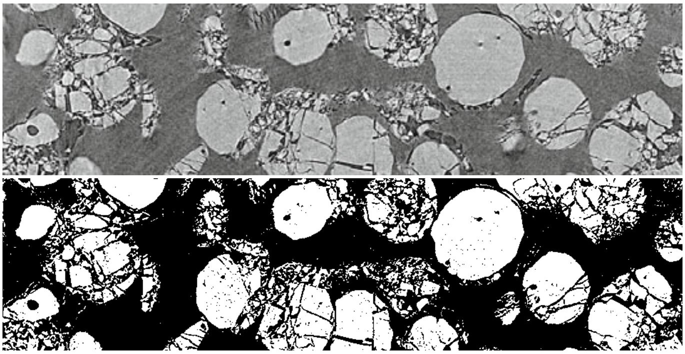


Fig. 8. Image E_1: a 3D X-ray microtomographic image of a fragmented granular material and its binarization by maximisation of the interclass variance (slice)

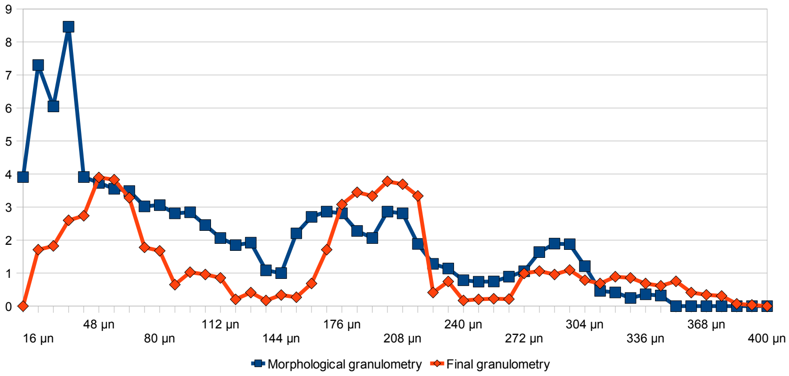


Fig. 9. Morphological granulometry of the X-ray microtomographic image of a fragmented granular material and granulometry of the final segmented image. On x axis, the diameter (μm), and on y axis the volume fraction of grains with this diameter.

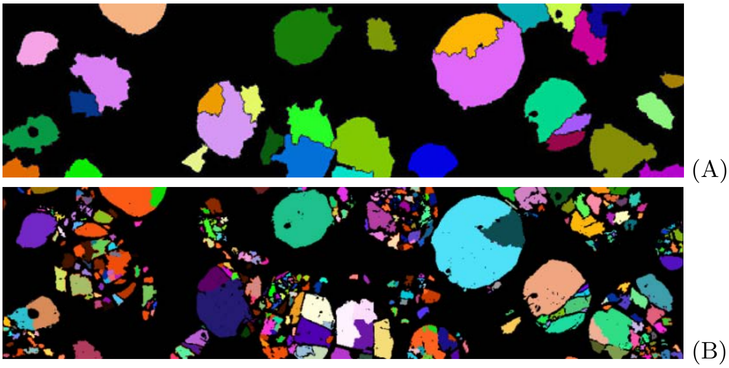


Fig. 10. Segmentation of the X-ray microtomographic image of a fragmented granular material (slice). (A) Results for the stochastic watershed approach. (B) Results for the granulometry-driven multiscale approach.

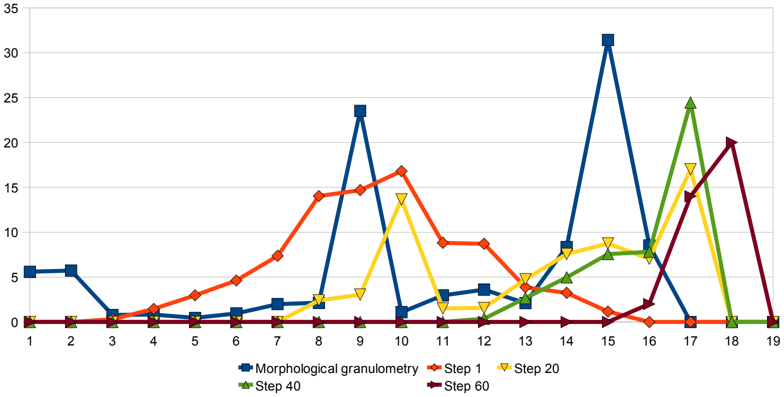


Fig. 11. Morphological granulometry of the random material R_1 and granulometry obtained at different steps of the hierarchy. The best result is step 40.

micro-CT. We are operating on a 954 × 243 × 243 subimage. It is a solid propellant sample with fragmented grains due to a mechanical impact. A voxel is 1.80μm. The original diameter of the grains is 400 μm, but there are many small fragments. A slice is shown on Fig. 8. Results of the segmentation are illustrated on Fig. 10 and Fig. 9, where it is clear that the multiscale approach overcomes the standard stochastic watershed algorithm.

5 Conclusion

On both the simulated image and the X-ray microtomographic image, our multiscale approach provides a good segmentation (it is essential to use a “volume” weighted granulometry, all our attempts with a ”number” granulometry being unsuccessful, due to its inherent sensitivity to noisy data). On the simulation, there is no over-segmentation for large grains, and the small grains are present as illustrated on Fig. 2 (C). On the microtomographic image, some of the large grains are over-segmented, but most of the small fragments are present and the results are better than the original stochastic watershed approach as illustrated on Fig. 10.

The granulometry of the segmented image fits to the morphological granulometry of the input images, since it was used as a constraint in the merging process (Fig. 12 and Fig. 9).

This new segmentation technique, combining an iterative probability-based merging of boundaries and a size distribution constraint, is very robust with respect to the noise contained in the image, without the necessity to apply a filter that would destroy the smallest grains. In addition, the segmentation is fully non-parametric, since at every step the required parameters are automatically deduced from the image.

Due to the graph approach, the process is much faster than the original stochastic watershed, as summarized on Tab. 1.

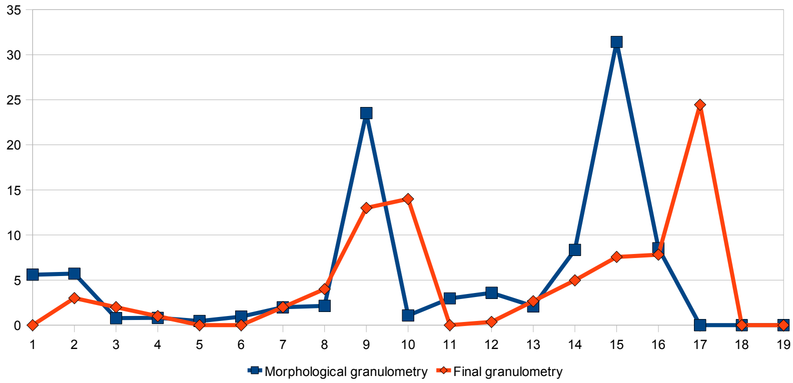


Fig. 12. Morphological granulometry of the random material R_1 and granulometry of the final segmented image

Table 1. Computational cost of the stochastic watershed, the graph-based stochastic watershed and our multiscale image segmentation approach. Times are give on a 3.00 GHz Pentium 4.

Algorithm	Image	Time
Stochastic Watershed	Random material R_1	1h 36m 36s
	Propellant (microtomographic)	3h 9m 21s
Graph-Based Stochastic Watershed	Random material R_1	5m 11s
	Propellant (microtomographic)	14m 9s
Multiscale Image Segmentation	Random material R_1	9m 22s
	Propellant (microtomographic)	25m 7s

Finally, it is possible to generalize the process to multi-criteria segmentation, using any other measurement tool as a constraint during the multiscale segmentation.

Acknowledgments. This work was supported by a grant fom DGA (contract 2009 34 0006). The authors are grateful to Alain Fanget (CEA Gramat) for his advice during this study.

References

1. Angulo, J., Jeulin, D.: Stochastic watershed segmentation. In: Proceedings of ISMM, 8th International Symposium on Mathematical Morphology, pp. 265–276 (2007) ISBN 978-85-17-00032-4
2. Beucher, S., Lantujoul, C.: Use of watersheds in contour detection. In: International Workshop on Image Processing, Real-Time Edge and Motion Detection (1979)
3. Beucher, S.: Watershed, hierarchical segmentation and waterfall algorithm. In: Proc. Mathematical Morphology and its Applications to Image Processing, Fontainebleau, pp. 69–76. Kluwer Ac. Publ., Nld. (1994)
4. Beucher, S., Marcotegui, B.: P algorithm, a dramatic enhancement of the waterfall transformation. CMM/Mines Paristech publication, 86 pages (September 2009)
5. Marcotegui, B., Beucher, S.: Fast implementation of waterfall based on graphs. In: Mathematical Morphology: 40 Years on: Proc. 7th ISMM, Paris, pp. 177–186. Springer, Heidelberg (2005)
6. Faessel, M., Jeulin, D.: Segmentation of 3D microtomographic images of granular materials with the stochastic watershed. Journal of Microscopy 239(1), 17–31 (2010)
7. Meyer, F., Stawiaski, J.: Tailored hierarchies for segmentation (submitted)
8. Jeulin, D.: Modèles morphologiques de structures aléatoires et de changement d'échelle. Thèse de Doctorat d'État, University of Caen, France (1991)
9. Jeulin, D.: Remarques sur la segmentation probabiliste, N-10/08/MM, Internal Report, Mines ParisTech. (September 2008)
10. Noyel, G., Angulo, J., Jeulin, D.: Random germs and stochastic watershed for unsupervised multispectral image segmentation. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part III. LNCS (LNAI), vol. 4694, pp. 17–24. Springer, Heidelberg (2007)

Pattern Recognition Using Morphological Class Distribution Functions and Classification Trees

Marcin Iwanowski and Michal Swiercz

Institute of Control and Industrial Electronics,
Warsaw University of Technology,
ul.Koszykowa 75, 00-662 Warszawa Poland
{iwanowski,swierczm}@ee.pw.edu.pl

Abstract. The paper presents an effective and robust method of classifying binary patterns. It starts with classification of foreground pixels of binary image into several spatial classes, which is performed using morphological image processing. By performing this classification with structuring elements of increasing sizes, the spatial class distribution functions are produced. These functions are normalized and sampled in order to obtain feature vectors of constant length that are invariant to pattern translation, rotation and scaling. Such feature vectors are next used to perform tree-based classification.

1 Introduction

In this paper, a method for recognizing binary patterns using morphological class distribution functions and decision trees is presented. The method is based on morphological classification. It allows extracting from the binary image pixels belonging to different *spatial classes* consisting of pixels characterized by particular morphological properties. Depending on the class being detected, various class extractors can be defined, based on morphological image processing operations. All the operators leading to extraction of spatial classes are using a single parameter – the structuring element of morphological operators. By applying structuring elements of increasing size when extracting spatial classes, *class distribution functions* can be obtained. They are expressing the dependence of the number or ratio of pixels belonging to a given spatial class on the size of the structuring element. The shape and characteristics of class distribution functions depend on the pattern for which they are computed. In the method presented in this paper, this function is normalized and sampled into a given number of samples using cubic spline interpolation. Thanks to this procedure, the scale-invariant feature vector of constant length is obtained. This trait, along with translation and (under some conditions) rotation invariance, make these features an effective tool for pattern recognition. In the proposed method, they are used as the input for tree-based classification. Classification trees, apart from their principal task – classification – allow also finding features that have a real influence on the classification result. An additional pre-processing of the training set is proposed in this paper – the feature preselection. By analysing the

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 143–154, 2011.

scattering measure for all pattern classes, features that are characterized by low in-class integrity are removed from this set. Training set with preselected features is finally used to train the classification tree. This tree is tested using a separate testing set.

The paper consists of 6 sections. In section 2, the idea and extraction method of morphological spatial classes are described. Section 3 presents the class distribution functions and their sampling. Section 4 describes the decision tree classification. In section 5, the test results are presented and finally, section 6 concludes the paper.

2 Morphological Spatial Classes

The classification task aims at assigning elements of the feature space into appropriate classes consisting of elements similar one to another. The the first application of mathematical morphology [6,7,9] to binary pattern classification was described in [5]. It was focused on forest analysis based on forest masks computed from remotely sensed images. Mathematical morphology tools were used to classify forest regions into 7 spatial classes. The applications of this methodology (called MSPA – morphological spatial patterns analysis) into forest pattern detection was also described in [3]. In [1,2,4] more generic view on this methodology was presented, allowing it to be applied to classify regions of various binary patterns: electronic circuit boards, water masks and binary shapes of various kinds.

```
......................  ......................
.11111...1111111111...  .bbbbb...cccccbbbbb...
.11111........1111111.  .baaab........baaabbc.
.11111111111111111111.  .baaabddddddddbaaabbc.
.11111........11111...  .baaab........baaab...
.11111..1.1111111111..  .baaab..e.ccccbaaabc..
.11111.11.....1111111.  .bbbbb.ee.....bbbbbcc.
......................  ......................
```

Fig. 1. Binary pattern (left) and five classes (right): a - *core*, b – *core boundary*, c – *branches*, d – *corridors*, e – *isolated*

The *morphological spatial class* of the binary image F is defined as a subset of foreground pixels characterized by a particular spatial property. Depending on the particular spatial class characteristics, various morphological operators should be applied to extract it. The only parameter used in the class extraction process is a structuring element B. In this paper we define the following set of spatial classes:

1. *Core* – region consisting of foreground pixels that are farther from the boundary of F than a distance implied by B. This class is obtained by means of erosion operator: $\Psi_{cr}(F, B) = F \ominus B$.

2. *Isolated* – connected components of the input image that do not contain any core pixel. This class is the residue of the morphological reconstruction of input image with core regions used as markers: $\Psi_{is}(F,B) = F \backslash F \triangle \Psi_{cr}(F,G)$.
3. *Core boundary* – region of pixels that are located inside the initial object that do not belong to the core region and are not farther from the core than the distance implied by B. This class can be obtained as a difference between opening and erosion: $\Psi_{cb}(F,B) = (F \circ B) \setminus (F \ominus B) = (F \circ B) \setminus \Psi_{cr}(F,B)$.
4. *Corridors* – groups of pixels which are neither core nor core boundary and which connect two disjoint core regions. A single object (connected component of foreground pixels) can contain more than just one core region. A connector between all cores of a single object that does not belong to core boundary is a corridor. Contrary to core boundary pixels, the corridor pixels are in a distance from the cores greater than implied by B. This class can be obtained by means of anchored homotopic skeletonization [6] of the input image with core pixels considered as anchor ones: $\Psi_{co}(F,B) = SKH\,(F, \Psi_{cb}(F,B))$, where $SKH(F,G)$ stands for the anchored homotopic skeleton of F with anchor pixels G.
5. *Branches* – groups of pixels which are neither core nor core boundary but are attached to a single core region (dead-ends of pattern): $\Psi_{br}(F,B) = F \setminus (\Psi_{is}(F,B) \cup \Psi_{cr}(F,B) \cup \Psi_{cb}(F,B) \cup \Psi_{co}(F,B))$.

The example of classification into five above classes for B equal to elementary 8-connected structuring element is shown on Fig. 2.

The result of classification strongly depends on the applied structuring element B – on its shape and size. These parameters imply the form of the pixel neighborhood considered. Consequently, they imply also the distance from a central pixel of the structuring element to other pixels belonging to it. Depending on the type of the structuring element various distance measures are considered. The elementary structuring elements induces either city-block distance (4-connected element) or max-norm distance (8-connected). Assuming that the distance from

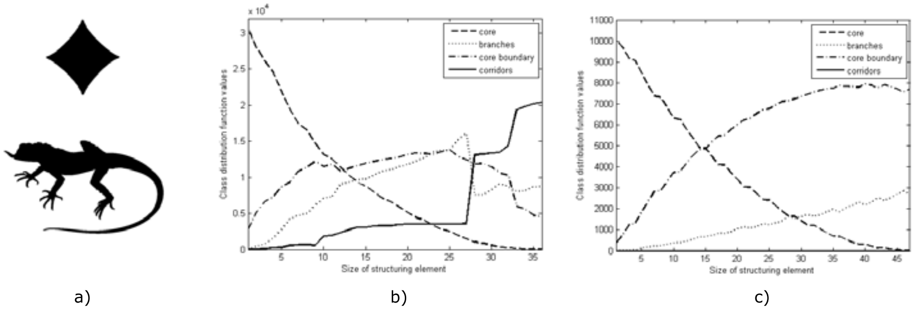


Fig. 2. Class distribution functions of *core*, *core boundary*, *branch* and *corridor* for test patterns: “lizard” (b) and “diamond” (c). Patterns are shown in (a).

pixels belonging to B is not grater than one, as in the above case, the notation $B^{(n)}$ will refer to a larger structuring element, which contains pixels not farther than n from the central pixel. $B^{(n)}$ is thus the neighborhood of radius n. The structuring element $B^{(n)}$ can be defined in various ways. The simplest (and fastest) is based on superposition by successive dilations of n elementary structuring elements $B^{(n)} = B \oplus B \oplus ... \oplus B$, where B stands for an elementary structuring element. In order to get the neighborhood of radius n according to the *Euclidean* distance the superposition by dilations cannot be used and the structuring element $B^{(n)}$ have to be computed individually for every n. Another possible choice of $B^{(n)}$ is the octagon-shaped element, that can be obtained by alternate usage of 4- and 8-connected elementary ones.

3 Class Distribution Functions and Feature Extraction

The results of spatial pixel classification depends on the size n of the structuring element $B^{(n)}$. Moreover this dependence differs from one binary pattern to another. The classification of image pixels using the series of structuring elements $B^{(n)}$ for increasing n allows obtaining the *class distribution functions* for each class. They are defined as:

$$\mathcal{D}_{CL}(n) = |\Psi_{CL}(F, B^{(n)})|, \tag{1}$$

where $|.|$ stands for the number of pixels of the argument and $CL \in \{cr, is, cb, co, br\}$ refers to the spatial class. Class distribution functions of four spatial classes (class *isolated* is not applicable in this case) of a test binary input pattern is presented in Fig. 2. It shows that functions of different shapes are noticeably different from one another.

The class distribution functions described above can be treated as an extension of granulometry. Granulometry by opening is equal to $\mathcal{D}_{cr}(n) + \mathcal{D}_{cb}(n)$, by opening by reconstruction to $\mathcal{D}_{cr}(n) + \mathcal{D}_{cb}(n) + \mathcal{D}_{co}(n) + \mathcal{D}_{br}(n)$.

The class distribution function defined by the Eq. 1 has some important properties. First, it depends on the form and size of the object(s) for which it is computed (see Fig. 2). This property makes it suitable for recognizing binary patterns. Second, it is always invariant to translations of objects within the image – this comes directly from the obvious property of morphological operators. Third, to some extent it is invariant to rotation. The extent depends on the type of the structuring element $B^{(n)}$. For elementary 4- and 8-connected elements it is invariant to rotations by $\frac{\pi}{2}$, in case of octagonal element – by $\frac{\pi}{4}$. In case of an Euclidean disk – by any angle. Finally, the class distribution function is not scale-invariant: scaling of the binary pattern implies scaling of the function defined by Eq. 1. This can be, however, countered by the normalization and sampling technique described further in this paper.

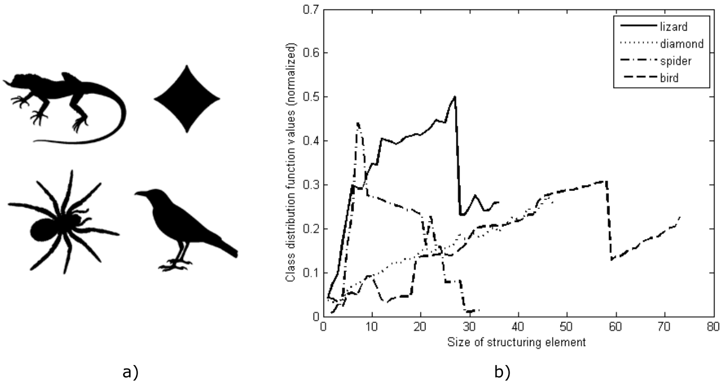


Fig. 3. Class distribution functions of class *branch (b)* for different test patterns *(a)*

In order to get scale-invariance, at first, the normalization along "y" axis is applied. It is obtained by dividing values of the function by a normalization factor equal to the total number of foreground pixels of the input image F:

$$\mathcal{D}'_{CL}(n) = \frac{|\Psi_{CL}(F, B^{(n)})|}{|F|} \tag{2}$$

In Fig. 3, normalized distribution functions of the class *branch* for various patterns are presented. It is worth noting at this point, that since class distribution functions depend only on the number of pixels of a certain class but not in any way on the position of these pixels within the image, it is possible to propose visually different shapes having highly similar class distribution functions (an example case is presented in Fig. 4). This very rare case, however, did not present itself during tests, in which shapes derived from physical objects were used.

Pattern scale influences also another parameter – the range of sizes of the structuring elements ("x" axis of the distribution function). The maximum effective value of n in Eq. 2 for which this function may change equals n_{MAX}, which is the largest size of erosion such that F does not disappear completely[1]. For all $n > n_{MAX}$ there are no more pixels belonging to the *core* class and all pixels are classified as *isolated*. For all $n \leq n_{MAX}$, at least two classes for each argument of the class distribution function always exist. Value n_{MAX} depends on the scale of a binary pattern. Assuming, for example, a binary pattern enlarged twice, one can observe that – comparing distribution functions of the original and the enlarged pattern – the latter has the same form as the former, but is stretched along the "x" axis and n_{MAX} is multiplied by 2. The values of a distribution function cannot be thus considered as scale-invariant features of patterns. Also, from the pattern recognition point of view, the feature vector

[1] In case of F foreground consisting of a single connected component, this is the size of erosion that produces ultimately eroded set.

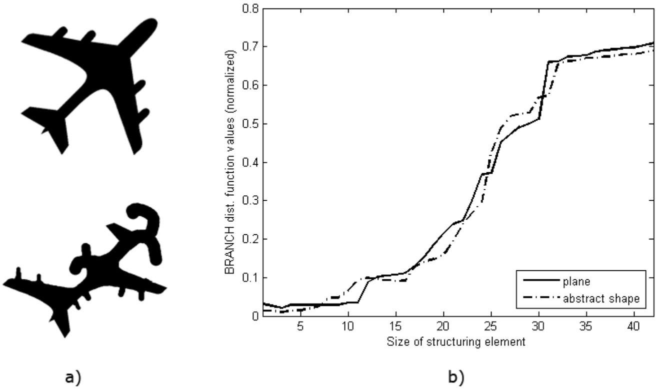


Fig. 4. Different shapes having a similar class distribution function: plane and abstract shape (a), and their "branch" class distribution functions (b)

consisting of features taken from the class distribution functions should have the same length independently from the range of structuring element sizes. This condition is however not fulfilled in the current case.

In order to produce a feature vector of the constant length, the class distribution function is sampled into a given number of samples s_{CL}. Due to the fact that sampling may require the values of the distribution function for real arguments, some interpolation is required. In the sampling method used in the experiments, the cubic spline interpolation was applied. As a result of sampling, the function defined by Eq. 2 of variable length is reduced to a given number of samples s_{CL}. The samples of distribution functions will be denoted using upper index in brackets and put together into a feature vector of the class: $\mathbf{v}_{CL} = \left[\mathcal{D}_{CL}^{(1)}, \mathcal{D}_{CL}^{(2)}, ..., \mathcal{D}_{CL}^{(s_{CL})}\right]$. An example of sampling is presented in Fig. 5. Two normalized class distribution functions of variants of the same pattern ("bird") are shown in a way that "x" axis was also normalized as: $n' = \frac{n}{n_{MAX}}$.

In order to get a complete morphological signature of the pattern, the feature vectors of all classes are grouped in a single feature vector of length $s = s_{cr} + s_{is} + s_{cb} + s_{co} + s_{br}$:

$$\mathbf{v} = [v^{(1)}, v^{(2)}, ..., v^{(s)}] = [\mathbf{v}_{cr}, \mathbf{v}_{is}, \mathbf{v}_{bc}, \mathbf{v}_{co}, \mathbf{v}_{br}] , \tag{3}$$

where $v^{(i)}$ stands for the i-th element of the feature vector (i-th feature, always $v^{(i)} \equiv \mathcal{D}_{CL}^{(j)}$ for certain j and CL). In case of patterns consisting of a single connected component, there is no need to use class *isolated* since it would always be empty. In such case, the feature vector of length $s = s_{cr} + s_{cb} + s_{co} + s_{br}$ is equal to:

$$\mathbf{v} = [v^{(1)}, v^{(2)}, ..., v^{(s)}] = [\mathbf{v}_{cr}, \mathbf{v}_{bc}, \mathbf{v}_{co}, \mathbf{v}_{br}] . \tag{4}$$

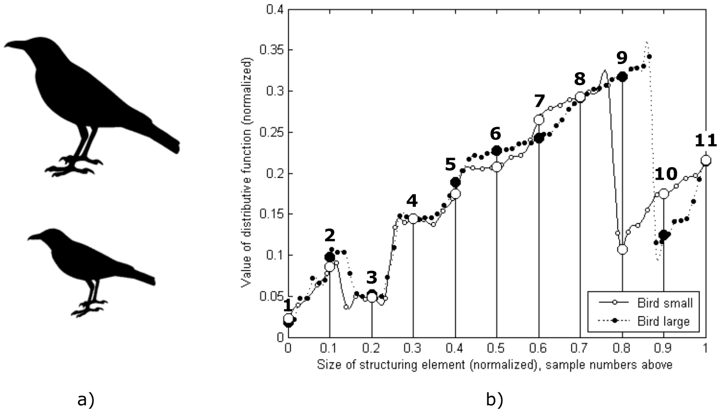


Fig. 5. Sampling of the *branch* class distribution functions *(b)* of two different variants of the "bird" pattern *(a)*

4 Tree-Based Classification

Classification (decision) trees [8] are graph structures where each node represents a certain decision rule, involving a test based on values of one of more features of the data set. These progressive tests divide the original dataset into disjoint subset nodes with higher class uniformity than the parent node. The final subsets which are not exposed to further divisions are called leaves and determine the class association of a case belonging to such a node. The number of leaves determines the tree size, while the number of edges between the root and the most distant leaves informs about the tree depth.

In order to perform the classification task, tree growing process is needed. It involves choosing the test conditions for each node, basing on a chosen quality criterion, in order to achieve the highest possible pattern class[2] uniformity in child nodes. The essential component of the tree-growing process is a training data set, i.e. a dataset, consisting of feature vectors of all pattern classes to be recognized. Moreover, each pattern type should be represented by multiple feature vectors computed for various pattern variations (also scaled, rotated, with disturbed boundary etc.). Let k be the number of all pattern classes. The set of feature vectors of all patterns in i-th pattern class is denoted as V_i. In fact this will be a matrix such that columns refer to features, while rows – to particular patterns of i-th type. In other words: $V_i = \left[\mathbf{v}_{i,1}^T, \mathbf{v}_{i,2}^T, ..., \mathbf{v}_{i,p_i}^T\right]^T$, where $\mathbf{v}_{i,j}$ stands for the feature vector (Eq. 3) of j-th example of i-th pattern type, p_i is the total number of examples of i-th pattern type in the training set and

[2] The notion of class is used in this paper in two meanings. *Spatial* class refer to the set of foreground pixles of the pattern (e.g. *core*, *branch*, etc.), while *pattern* class – to type of the pattern (e.g. "bird","spider", etc.).

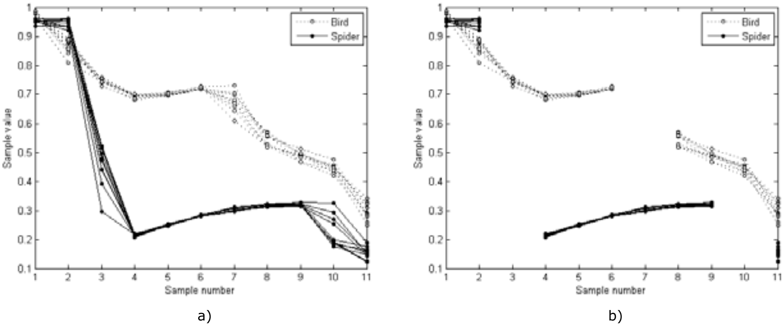


Fig. 6. Sampled features, class *core boundary* ($\mathcal{D}_{cb}^{(1)}, ..., \mathcal{D}_{cb}^{(11)}$) – without feature preselection (a), with feature preselction (b)

the upper index T stands for vector transposition. The whole training set is thus defined as $V = \{V_1, V_2, ..., V_k\}$.

As it was pointed out earlier, normalized and sampled class distribution functions, in general, are fairly invariant to rotation and scaling. However, under certain conditions (for example, when using a structuring element that is highly asymmetrical in respect to rotations), some features can show a level of undesired in-class scattering and it is necessary to remove them from the feature vectors, so that they are not used in the training of the decision tree. We call this process *feature preselection.* As a measure of scattering, we use standard deviation. A global threshold t is chosen to test it. Features with a standard deviation exceeding this threshold for at least one pattern type are removed from the feature set. In other words, in further processing, only features of index l that fulfill the below condition are kept:

$$\sqrt{\frac{1}{p_i}\sum_{j=1}^{p_i}\left(v_{i,j}^{(l)} - m_i^{(l)}\right)^2} < t \ , \ \ \forall i = 1, ..., k \ , \tag{5}$$

where $v_{i,j}^{(l)}$ stands for l-th feature of the j-th example of i-th pattern class, $m_i^{(l)}$ is the mean value of l-th feature in i-th pattern class. This guarantees that features used in the decision tree learning, chosen as split points for the decision tree present high in-class integrity and the classification result will not be influenced by common, slight distortions of the processed patterns. This is demonstrated in Fig. 6 showing two different shape classes, on features derived from the core boundary pixel class. Samples $\mathcal{D}_{cb}^{(3)}$ and $\mathcal{D}_{cb}^{(10)}$ were removed from the feature set due to excess scattering for shape "spider", and sample $\mathcal{D}_{cb}^{(7)}$ was removed due to excess scattering for shape "bird". Therefore, the following subset of the initial samples of spatial class *core boundary* can be used in the decision tree growing: $\{\mathcal{D}_{cb}^{(1)}, \mathcal{D}_{cb}^{(2)}, \mathcal{D}_{cb}^{(4)}, \mathcal{D}_{cb}^{(5)}, \mathcal{D}_{cb}^{(6)}, \mathcal{D}_{cb}^{(8)}, \mathcal{D}_{cb}^{(9)}, \mathcal{D}_{cb}^{(11)}\}$. During the tree growing process, the split criteria are chosen for each node to maximize class integrity in child nodes after the division. There exist, however, some severe problems in decission tree learning, such as *overfitting,* which can cause the tree to become overly complicated and unflexible. This phenomenon arises from the discrete and inherently greedy nature of the tree-growing algorithms, trying to properly include every data point into the tree structure, even if it is a statistically insignificant outlier. In our method, due to feature preselection and elimination of the most likely error-causing and outlier-influenced sections of the feature set, tree overfitting does not present itself as a problem, so it is possible to obtain full, unpruned trees with completely class-uniform leaves.

5 Results

Two rounds of testing were performed to evaluate the quality of classification. In the first round, a classification tree was constructed to perform recognition of binary shapes of 4 distinct pattern classes: “lizard”, “diamond”, “spider”, “bird” (shown in Fig. 3(a)). A training set of 160 binary patterns was used, where each pattern class was represented by 40 examples. The diversity of images within each pattern class was high, as shown in Fig. 7. It is worth pointing out, that the shapes differ from each other quite noticeably (compare the shape of the tail between (a) and (c)). Morphological classification, distribution function normalization and sampling were preformed, with $s_{CL} = 11$ samples for every spatial class CL. Since all patterns are represented by single connected components, class *isolated* was not taken into account – the feature vector was created based on the Eq. 4. The number of samples equal to 11 was enough to maintain the main characteristics of the class distribution function in the feature set at a satisfactory level. The preselection threshold for this round of testing was set to 0.035.

The tree-growing algorithm generated an output tree of size 4 and depth 3. This tree is shown in Fig. 9. The division points were chosen as samples $v^{(6)}$, $v^{(13)}$, and $v^{(38)}$. The samples forming the feature vector $\mathbf{v}$ are organized in the following manner (spatial classes and samples): *core* - $v^{(1)},...,v^{(11)}$, *branches* - $v^{(12)},...,v^{(22)}$, *core boundary* - $v^{(23)},...,v^{(33)}$, *corridors* - $v^{(34)},...,v^{(44)}$. Therefore, samples belonging to spatial classes: *core* ($v^{(6)} \equiv \mathcal{D}_{cr}^{(6)}$), *branch* ($v^{(13)} \equiv \mathcal{D}_{br}^{(2)}$) and *corridor* ($v^{(38)} \equiv \mathcal{D}_{co}^{(5)}$) were chosen as decision tree splits. Tree testing was performed to verify the quality of the classifier. The test set consisted of 80 shapes, 20 for each of the shape types present in the training set

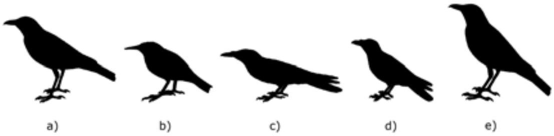


Fig. 7. Sample shapes for feature set generation: “bird1” *(a)*, “bird2” *(b)*, distorted *(c)* and undistorted *(d)* shape “bird3”, distorted “bird1” *(e)*

("lizard","diamond","spider","bird"). The images were chosen basing on the same criteria as with the training set: there were different natural shapes and shapes derived from these natural shapes by means of scaling and rotation. This set was tested against the obtained classification tree, and a 100% classification accuracy was achieved. In this test scenario, the best class predictors turned out to be the *core*, *corridors* and *branch* spatial classes. The distribution functions for these pixel classes show a high diversity between the shapes, and their characteristics are highly distinctive. It is therefore feasible to perform a pre-analysis on smaller sets and construct a feature set consisting only of a subset of pixel class samples to achieve good classification results for a specific pattern composition.

A second test was performed on a larger set of shapes. This time patterns was belonging to 8 classes: "bird", "spider", "diamond", "lizard", "plane", "octopus", "hand" and "whale" (shown in Fig. 8).

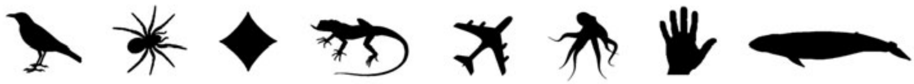


Fig. 8. Sample shapes used for the second round of testing

The training set consisted of 320 images, where each class was represented by 40 images derived from the basic shape by means of scaling and rotation to achieve high in-class diversity. The test set consisted of 160 images, with 20 images representing each of the 8 shape classes. Distribution function calculation, feature extraction, preselection and tree growing were performed as in the first round of testing, with $s_{CL} = 11$ samples for every spatial class CL. The preselection threshold was set to 0.135. Again, high accuracy of classification was achieved, with 98.75% of shapes being properly attributed to their shape class (two objects were improperly classified). The decision tree generated in the second round of testing is presented in Fig. 10. This time, samples belonging to the following spatial classes were used as decision tree splits: *core* ($v^{(1)} \equiv \mathcal{D}^{(1)}_{cr}$,

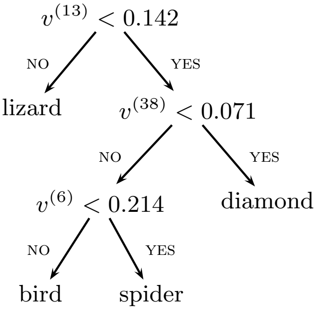


Fig. 9. Decision tree structure with marked split points

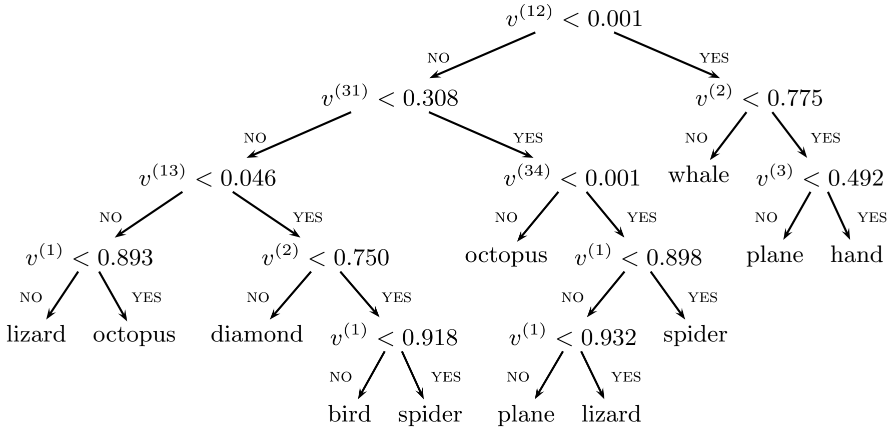


Fig. 10. Decision tree structure generated in the second round of testing

$v^{(2)} \equiv \mathcal{D}_{cr}^{(2)}$, $v^{(3)} \equiv \mathcal{D}_{cr}^{(3)}$), *branch* ($v^{(12)} \equiv \mathcal{D}_{br}^{(1)}$, $v^{(13)} \equiv \mathcal{D}_{br}^{(2)}$), *core boundary* ($v^{(31)} \equiv \mathcal{D}_{cb}^{(9)}$) and *corridor* ($v^{(34)} \equiv \mathcal{D}_{co}^{(1)}$). Similarly to the first test, in the second test scenario the shape classification was determined most heavily by the *core* and *branch* spatial class distributions.

6 Conclusions

In the paper, the method for classifying binary patterns was proposed. It consists of several steps and starts with classification of pixels belonging to binary patterns into several spatial classes, which is performed using morphological image processing. By performing this classification with structuring elements of increasing sizes, the spatial class distribution functions are produced. These functions are normalized and sampled in order to obtain feature vectors of constant length that are invariant to translation, rotation and scaling of binary pattern. Such feature vectors are next used to perform decision-tree classification. Prior to proper decision-tree classification an additional step of feature preselection is performed based on the training set. It allows removing from this feature set features with high intra-class scattering, and in effect, makes the remaining features more suitable for pattern class separation.

The tests confirm that the proposed method is a robust and effective tool for binary pattern recognition. Furthermore, the method shows a level of flexibility and, if required, can be optimized for a specific set of shapes to achieve better performance and match accuracy.

As pixel class extraction from pictures containing the binary shapes is the most time-consuming step of the classification task, in most cases it is possible to narrow the analysis to a subset of pixel classes (for example, in the test scenario presented in this paper, only three pixel classes proved relevant). Moreover, since pixel class extraction is a task that is limited in dependencies to the individual picture processed, it can be performed in parallel by multiple instances of the extracting program, running on a multi-core CPU or multi-processor machine. Morphological operations performed during feature extraction can also be parallelized to a large extent, allowing for further reduction of processing time on modern computer hardware.

References

1. Soille, P., Vogt, P.: Morphological segmentation of binary patterns. Pattern Recognition Letters 30 (2009)
2. Iwanowski, M.: Morphological Classification of Binary Image Pixels. Machine Graphics and Vision 18 (2009)
3. Riitters, K., Vogt, P., Soille, P., Kozak, J., Estreguil, C.: Neutral model analysis of landscape patterns from mathematical morphology. Landscape Ecology 22, 1033–1043 (2007)
4. Iwanowski, M.: Binary Shape Characterization using Morphological Boundary Class Distribution Functions. In: Advances in Intelligent and Soft Computing - Computer Recognition Systems 2. Springer, Heidelberg (2007)
5. Vogt, P., Riitters, K., Iwanowski, M., Estreguil, C., Kozak, J., Soille, P.: Mapping Landscape Corridors. Ecological Indicators 7(2) (2007)
6. Soille, P.: Morphological image analysis. Springer, Heidelberg (1999) (2004)
7. Serra, J., Vincent, L.: An overview of morphological filtering. Circuit Systems Signal Processing 11(1) (1992)
8. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. CRC Press, Boca Raton (1984)
9. Serra, J.: Image analysis and mathematical morphology, vol. 1. Academic Press, London (1983)

Object Descriptors Based on a List of Rectangles: Method and Algorithm

Marc Van Droogenbroeck and Sébastien Piérard

INTELSIG Laboratory, Montefiore Institute, University of Liège, Belgium

Abstract. Most morphological operators use a unique structuring element, possibly at different scales, to describe an object. In addition, morphological algorithms are often restricted to 1D structuring elements, combinations of 1D elements, or isotropic structuring elements (like circles), because of the lack of methods directly applicable to arbitrary shaped 2D structuring elements. While these descriptors have proved useful in the past, we propose an alternative that uses the list of maximal rectangles contained in a set X.

In particular, we focus on an opening that preserves large rectangles contained in a set X and on its companion 2D algorithm that builds a list of all the maximal rectangles that fit inside an arbitrary set X. This list is the base of new descriptors that have been used successfully for machine learning tasks related to the analysis of human silhouettes.

For convenience, we provide the C source code and a program of our algorithm at `http://www.ulg.ac.be/telecom/rectangles`

Keywords: Opening, Granulometry, Algorithm, Rectangle.

1 Introduction

Tools for describing the shape of an object are useful for many applications, including classification. In mathematical morphology, the numerous tools include erosions, openings, skeletons, distance functions, etc. In that context, opening and closing operators play an important role, mainly because of their useful property of idempotence which is similar to the notion of ideal filter in linear filtering. If the property and behavior of many openings (like morphological openings [7], area openings [9], openings by reconstruction [6], or attribute openings [4]) are well known to practitioners, their implementation might still be problematic, mainly for multidimensional spaces. For example in the particular case of openings with rectangles, it is common to decompose the structuring element as the dilation of an horizontal line by a vertical line, and to apply the chain rule. While this procedure is extendable to 3D objects, algorithms that rely on a decomposition impose a processing order and become less convenient for 2D granulometries because intermediate results have to be stored.

From a practical point of view, we could classify binary openings in two families: (1) the family of openings that compare a structuring element to the set X to be interpreted (these openings are called morphological openings hereafter), and

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 155–165, 2011.

(2) the family of attribute openings. Unlike morphological openings, attribute openings preserve the shape of a set X, because they simply test whether or not a connected component satisfies some increasing criterion Γ, called *attribute*. An example of valid attribute consists in preserving a set X if its area is superior to λ or removing it otherwise. This is in fact the surface area opening [9]. More formally, the attribute opening γ_Γ of a connected set X preserves this set if it satisfies the criterion Γ:

$$\gamma_\Gamma(X) = \begin{cases} X, & \text{if } X \text{ satisfies } \Gamma, \\ \emptyset, & \text{otherwise.} \end{cases} \tag{1}$$

Morphological openings affect the shape of an object. Therefore, in order to build a more descriptive shape analysis tool, they are often characterized by a parameter k leading to granulometries (with some specific restrictions on the shape of the structuring element) and granulometric functions or curves that provide a numeric result increasing (or decreasing) with k. Likewise, if attribute openings do not affect the shape of regions that are preserved (because they preserve the entire connected component [4]), they can be parametrized to provide a granulometric curve too. The underlying limitations are that it is hard to build a two-dimensional granulometric function and that all information about the location of an object in the image is lost with a granulometric curve.

In [8], we proposed an algorithm that computes a list of rectangles and derive granulometric curves. The advantages of having a list of rectangles are twofold: (1) it is simple to calculate granulometric curves once the list has been built because the list gathers all the information needed for the granulometry, and (2) it is possible to calculate, for each pixel, some statistics extracted from the list, like the size of the largest rectangles comprising that pixel. Descriptors based on a list of rectangles have proved successful for two classification tasks related to the analysis of human activities; they were used for gait recognition in [3], and for identifying human silhouettes in video scenes both in 2D [2] and in 3D [5].

In the following section, we illustrate our approach with one possible operator computable directly from a list of rectangles and show how to use it to build a granulometric curve. In Section 3, we describe an algorithm that computes the list of rectangles contained in an arbitrarily shaped object X. Section 4 concludes the paper.

2 Towards a Family Opening

2.1 Reminder

Consider the discrete space $\mathbb{Z}^2$. Given a set $X \subseteq \mathbb{Z}^2$ and a vector $b \in \mathbb{Z}^2$, the translate X_b is defined as $X_b = \{x + b | \, x \in X\}$. Let us take two subsets X and B of $\mathbb{Z}^2$. The dilation and erosion are respectively defined as

$$X \oplus B = \bigcup_{b \in B} X_b = \bigcup_{x \in X} B_x = \{x + b | \, x \in X, \, b \in B\}, \tag{2}$$

$$X \ominus B = \bigcap_{b \in B} X_{-b} = \{p \in \mathbb{Z}^2 | \, B_p \subseteq X\}. \tag{3}$$

Dilation and erosion are not inverse operators. If X is eroded by B and then dilated by B, one may end up with a smaller set than the original set X. This set, denoted by $X \circ B$, is called the *opening* of X by B and defined by $X \circ B = (X \ominus B) \oplus B$. One useful property of openings is that a morphological opening is the union of all the translate B_p included in X, that is

$$X \circ B = \bigcup \{B_p \,|\, B_p \subseteq X\}\,. \tag{4}$$

2.2 Definition of a Family Opening

We define a parametric opening operator that encompasses all the rectangles whose cardinality is larger or equal to k. In the following, we limit possible rectangles to rectangles whose sides are parallel to the $x - y$ system coordinates; in other words, all rectangles can be expressed as $nH \oplus mV$, where H and V are horizontal and vertical segments respectively. We could also consider other directions but the algorithm described in Section 3 then would have to be adapted.

The *family opening* of a set X by a family of rectangles whose cardinality or area is larger or equal to k, denoted $\gamma_k(X)$ hereafter, is defined by

$$\gamma_k(X) = \bigcup \{R \,|\, \#(R) \geq k \text{ and } R \subseteq X\}\,, \tag{5}$$

where $\#(R)$ denotes the cardinality of a rectangle R. This operator is the union of openings by all the rectangles that meet the size constraint, $\#(R) \geq k$, therefore it is an opening , but it is not an area opening as not only the area but also the shape is constrained. Note that we could use a different criterion to select rectangles from the list. In [3] for example, the shape descriptor is based on the histograms of all the rectangle widths and heights. A subset of all maximal rectangles was successively used to discriminate human shapes from objects shapes in [2].

A granulometric curve is easily derived from $\gamma_k(X)$. Granulometric curves can be obtained by taking the cardinality of the reconstructed area with all the rectangles that have a cardinality larger than a given threshold, which is the area of $\gamma_k(X)$. Fig. 1 draws these granulometric curves for some simple binary shapes. To ease the interpretation, we have only displayed the values for area threshold when they change the cardinality of the reconstructed area (we have removed all the plateaus); it is also possible to interpolate the values to smooth the curves. It is interesting to note for example that the overall shape of the granulometric curves of a diamond and a circle are similar, but that gaps are larger for a circle. Classification tools and machine learning techniques are adequate to interpret the information provided by such granulometric curves, for example to differentiate between several shapes.

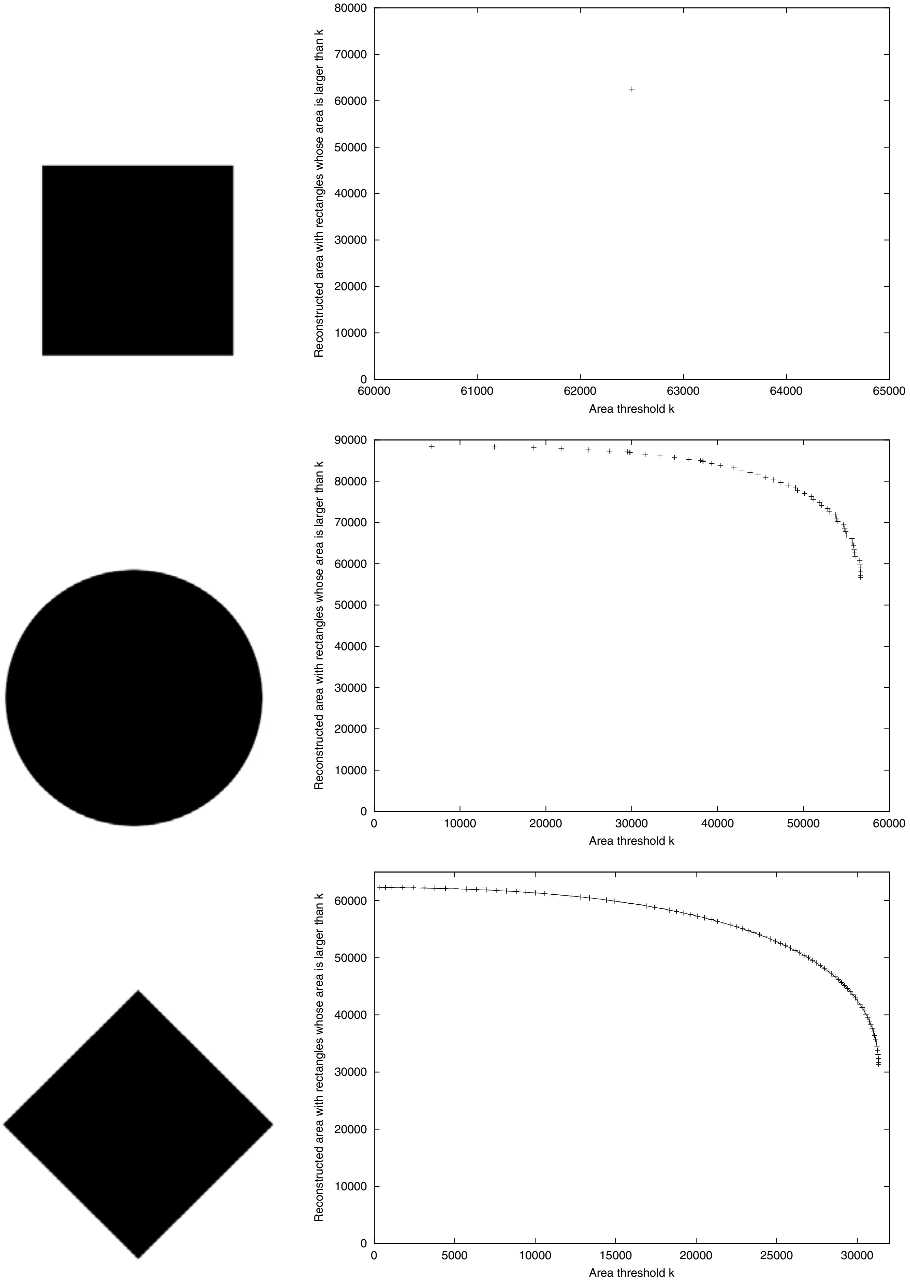


Fig. 1. Simple binary objects and their corresponding granulometric curves (obtained by taking the area of the family opening $\gamma_k(X)$ with respect to k)

3 Description of an Algorithm That Builds a List of All Maximal Rectangles Contained in a Binary Set X

While Vincent [10] proposed an efficient algorithm for computing 1D granulometries, only a few algorithms are applicable to 2D spaces. Several authors proposed variants that rely on the chain rule, which states that $X \ominus (H \oplus V) = (X \ominus H) \ominus V$ and that $X \oplus (H \oplus V) = (X \oplus H) \oplus V$, to deal with rectangles, but we consider that these algorithms are 1D in nature. For example, some decomposition properties were used by Bagdanov and Worring in [1] to derive rectangular granulometries and interpret the similarity of document images. In [8], we proposed a complex algorithm that computes two intermediate images containing all the necessary information for granulometries by rectangles. While this algorithm is efficient, the new algorithm described hereafter is much simpler and fast enough for real-time applications.

3.1 The Principles

There are different ways to characterize rectangles and to build a list of rectangles that fit inside an object X. In the following, we concentrate on maximal rectangles. By definition, if R is a maximal rectangle, then there is no R' such that $R \subset R' \subseteq X$. Note that if we consider all the possible rectangles, including the singleton (a rectangle degenerate to a single pixel), then for each location $x \in X$, there is at least one maximal rectangle that contains x. Of course, there might be more such rectangles.

A first subtle difference with the algorithm proposed in [8] is that we do not impose that each rectangle contains at least one point not included in any other rectangle of the list (this would lead to a minimal cover by rectangles). A second difference is the way to find the rectangles and the association of a rectangle with a *reference point*.

Before we define the notion of reference point, let us first remark that each maximal rectangle touches the upper, lower, left, and right borders; otherwise the rectangle would not be maximal. Our objective is to associate a unique reference point for each rectangle, but we accept that rectangles share a common reference point. By convention, we choose the reference point of a maximal rectangle to be the lowest point of the rectangle that touches the left border of the object.

Our algorithm is built around the concept of reference point. But, because we are not able to localize reference point in advance in an image, we introduce another notion, which is that of *candidate*. Candidates of a set X are elements of X that might be reference points of a maximal rectangle included in X. By definition, candidates are located on the left border of the object. This does not mean that all these points are reference points, but at least they are candidates.

The steps of our algorithm are then:

1. determine candidates, that is possible reference points,
2. search all the rectangles that could be associated to a candidate,
3. associate the maximal rectangle to a reference point if such a rectangle exists, and add the rectangle to the list of maximal rectangles.

A candidate is first selected during a scanning process of the image. A specific rule applies to the detection of candidates. If a maximal rectangle contains two candidates in the same column, the highest of the two candidates has to be ignored because that candidate is not a reference point according to our definition. Consequently, the downwards extension of a maximal rectangle rectangle starting from a reference point is bounded by the next to left contour point located beneath the reference point; this downwards vertical extension value is denoted maxS hereafter (see Fig. 2 for a graphical illustration of maxS). This specific rule eases the search for maximal rectangles.

The steps of our algorithm are illustrated on Fig. 2. The candidate is represented by a dark disk. We have put a horizontal line to denote the downwards limit maxS. The respective drawings are described hereafter.

1. Suppose that we have detected a candidate located at $(\mathrm{col},\ \mathrm{row})$. We consider a first rectangle whose width is 1 and height is maximal. As the rectangle is extendable to the right with the same height, this rectangle is discarded.
2. The first rectangle is then extended to the right to reach a width of 2. This rectangle is maximal because it does not extend to the right, but it crosses the lower limit as defined by maxS. We have to ignore this rectangle too because the candidate is not the reference point for this rectangle. Depending on the scanning order, this rectangle is discovered earlier (upwards column scanning) or later (downwards column scanning).
3. We reduce the height of the rectangle and extend it to the right. This rectangle is maximal and its downwards extension is inferior or equal to maxS. Therefore the candidate is indeed a reference point and we must add this rectangle to the list.
4. The height of the third rectangle is reduced and the rectangle is extended to the right. This rectangle is not maximal, therefore, it is also discarded.
5. Again, we extend the rectangle to the right. The resulting rectangle is maximal and does not cross the lower limit. It is therefore added to the list of maximal rectangles.
6. Finally, we reduce the height of the fifth rectangle and extend it to the right. Since this sixth rectangle touches the right border of the object on the row row, it is maximal. Furthermore, it does not cross the lower limit. It is thus also added to the list.

Note that we now have three maximal rectangles associated to the same reference point located at $(\mathrm{col},\ \mathrm{row})$. Therefore, the number of reference points is not an upper bound of the number of maximal rectangles.

3.2 Implementation

Listing 1 provides a C implementation of our algorithm. Note that 50 lines of C code suffice to compute the list! The function “listRectangles” computes the list of rectangles for an object represented by a 2D array of booleans named “object”. This array has w columns and h rows. For each discovered maximal rectangle,

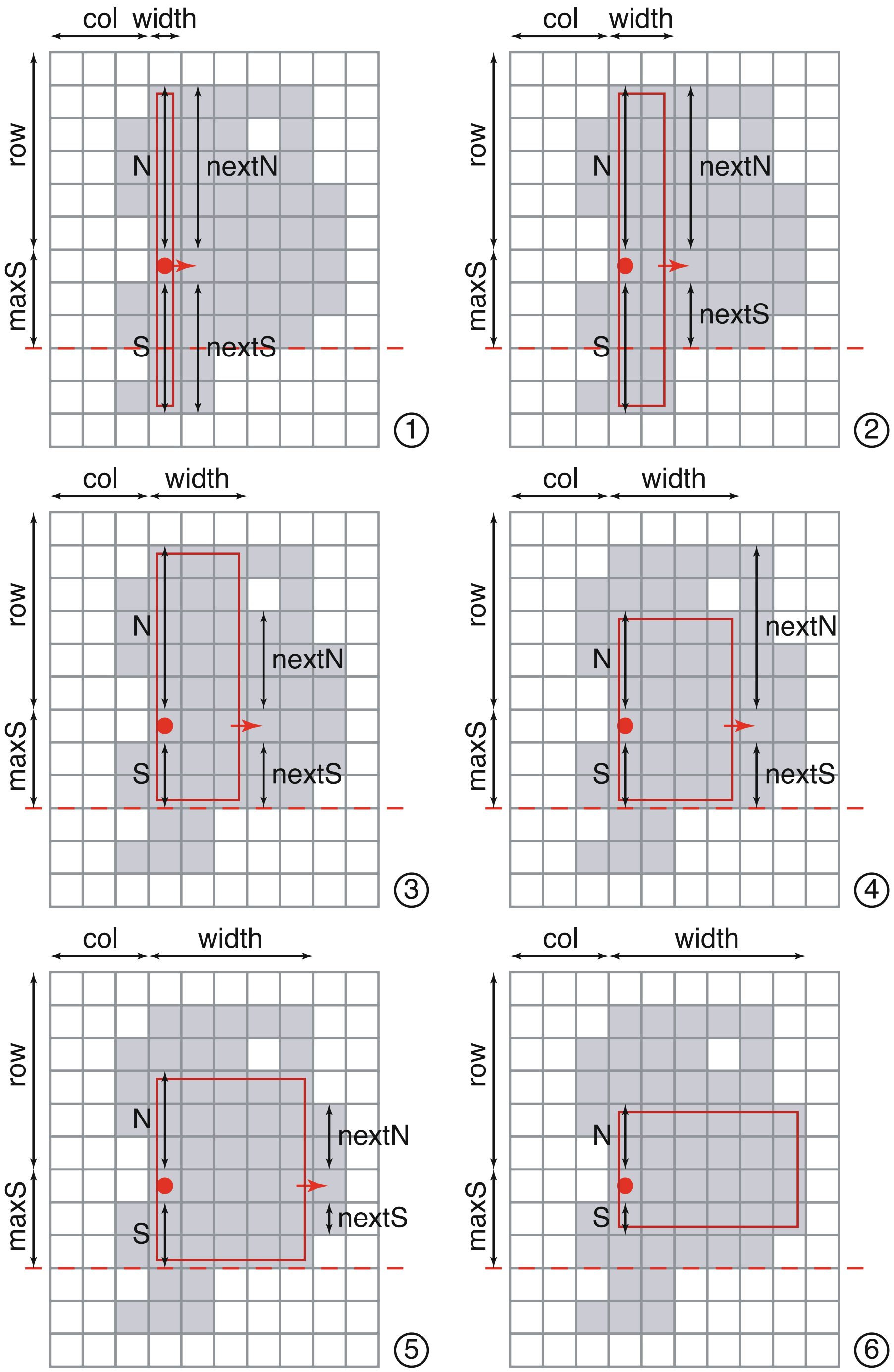


Fig. 2. Steps to detect all the rectangles associated to a candidate (denoted by a disk on the drawings)

Algorithm 1. An algorithm that detects all the maximal rectangles contained in an object X represented by a binary image named "object [w][h]".

```
void listRectangles ( int w , int h , bool object [ w ][ h ] ) {

  int dN [ w ][ h ] ;
  for ( int col = 0 ; col < w ; ++ col ) {
    dN [ col ][ 0 ] = object [ col ][ 0 ] ? 0 : -1 ;
  }
  for ( int row = 1 ; row < h ; ++ row ) {
    for ( int col = 0 ; col < w ; ++ col ) {
      if ( ! object [ col ][ row ] ) dN [ col ][ row ] = -1 ;
      else dN [ col ][ row ] = dN [ col ][ row - 1 ] + 1 ;
    }
  }

  int dS [ w ][ h ] ;
  for ( int col = 0 ; col < w ; ++ col ) {
    dS [ col ][ h - 1 ] = object [ col ][ h - 1 ] ? 0 : -1 ;
  }
  for ( int row = h - 2 ; row >= 0 ; -- row ) {
    for ( int col = 0 ; col < w ; ++ col ) {
      if ( ! object [ col ][ row ] ) dS [ col ][ row ] = -1 ;
      else dS [ col ][ row ] = dS [ col ][ row + 1 ] + 1 ;
    }
  }

  for ( int col = w - 1 ; col >= 0 ; -- col ) {
    int maxS = h ;
    for ( int row = h - 1 ; row >= 0 ; -- row ) {
      ++ maxS ;
      if ( object [ col ][ row ]
            && ( col == 0 || ! object [ col - 1 ][ row ] ) ) {
        int N = dN [ col ][ row ] ;
        int S = dS [ col ][ row ] ;
        int width = 1 ;
        while ( col + width < w && object [ col + width ][ row ] ) {
          int nextN = dN [ col + width ][ row ] ;
          int nextS = dS [ col + width ][ row ] ;
          if ( ( nextN < N ) | ( nextS < S ) ) {
            if ( S < maxS ) add ( col , row - N , width , N + S + 1 ) ;
            if ( nextN < N ) N = nextN ;
            if ( nextS < S ) S = nextS ;
          }
          ++ width ;
        }
        if ( S < maxS ) add ( col , row - N , width , N + S + 1 ) ;
        maxS = 0 ;
      }
    }
  }

}
```

the function calls a callback function named "add" with four arguments: the top left corner coordinates, the width, and the height. Note that, in this code, we have arbitrarily chosen the column-major order.

Some parts of the C code are commented hereafter.

- Lines 3 to 12. To optimize the search, we first compute the distance that separates a pixel located at (col , row) to the upper border of the object. The result is stored in the dN [col][row] data structure; the dN notation stands for *Distance to the North*.
- Lines 14 to 23. Likewise, distances between (col , row) and the downwards border is computed and stored in a specific data structure dS [col][row]. About half of the code is devoted to these simple search operations!
- Lines 25 and 27. This is the main loop on all the pixels and, inside the loop, the current location is (col , row).
  Ideally, only the border should be examined because reference points have to belong to the border, but this supposes that the border of the object is known prior to the scanning process. Note that the image is scanned upwards (line 27). This is an indirect consequence of our definition for reference points, because this order eases the determination of maxS, as explained hereafter.
- Lines 29 to 30. We detect if a pixel is a candidate. Only pixels located on the left border are candidates.
- Lines 31 to 44. All the maximal rectangles comprising (col , row) are considered. The upper left corner of such a maximal rectangle, its width, and its height are respectively (col , row − N), width, and N + S + 1.
- Lines 38 and 44. A rectangle is added to the list only if (col , row) is a reference point. An efficient way to determine if a candidate is a reference point uses the instructions of lines 26, 28, and 45. The scanning order helps us to determine the distance between (col , row) and a reference point located downwards in the same column, that is (col , row + maxS). This distance is stored in a variable called maxS, which is initialized as if a previous virtual reference point was located outside the image (therefore maxS is taken such that it is equal to h which is a value larger than the possible values, at line 26). Indeed, (col , row + h) is located outside the image. Once a candidate has been dealt with and before the algorithm moves to the next row index upwards, maxS is set to 0 (line 45) which avoids redundant rectangles in the list.

In the next two subsections, we discuss two issues related to the algorithm: the maximal size of the rectangle list, in order to bound the memory needed to store the list, and the complexity of the algorithm.

3.3 On the Number of Maximal Rectangles

We have seen that the number of reference points is not directly useful to determine a bound for the number of maximal rectangles contained in a binary set X. However, for each maximal rectangle, the location (col + width − 1 , row) plays a particular role. It is located on the same row as the reference point and it indicates the right edge of the rectangle. By construction, it is impossible for two maximal rectangles to share this point. This observation allows us to derive an important upper bound for the number of rectangles on an object.

Property 1. The number of maximal rectangles contained in an object X is bounded by the cardinality of X.

Consequently, the maximal memory footprint to store the list of rectangles is bounded by storage size of one rectangle multiplied by the cardinality of an object X, which in some cases might be as large as the image.

3.4 Run-Time Complexity Analysis

Let $w \times h$ be the size of the image, and A the cardinality (that is the area expressed in pixels for discrete sets) of the object X. The overall run-time complexity of our algorithm is $\mathcal{O}(wh)$. Here are the details:

- The processing cost for adding rectangles to the list (which is expressed in the code by calling the function "add") is $\mathcal{O}(A)$. This is derived from Property 1.
- Precomputing dN and dS for each point inside X has a complexity of $\mathcal{O}(wh)$.
- Likewise, detecting all candidates also takes $\mathcal{O}(wh)$.
- For each candidate and each value of width to be considered, deciding if the rectangle is maximal plus computing the location and the size of this rectangle is achieved in constant time. Therefore, if dE is the distance between the candidate and the right border of X, the complexity of these operations is given by $\mathcal{O}(dE)$ for each candidate. As a conclusion, once the candidates are known, computing the list of rectangles only takes $\mathcal{O}(A)$, where A is the cardinality of X.

The combination of the complexity of all these steps leads to a complexity of our algorithm given as $\mathcal{O}(wh)$. In practice and in our algorithm, the bottleneck originates from the computation of dN and dS, and the detection of all candidates.

4 Conclusions

This paper presents methods that deal with rectangles to characterize the shape of objects. These methods, that rely on the list of all the maximal rectangles included into an arbitrarily shaped object X, offer an interesting alternative to 1D or isotropic descriptors. In addition, we propose an efficient algorithm that computes this list. For convenience, we provide the C source code and a program of our algorithm at http://www.ulg.ac.be/telecom/rectangles

From this list, it is possible to derive granulometric curves and many other statistics. Some of these statistics were used successfully for gait recognition and for the detection of human silhouettes. New opportunities that originates from the possibility to determine local statistics (that is, statistics for each pixel) are open for future works.

Acknowledgments. S. Piérard has a grant funded by the FRIA (http://www.frs-fnrs.be/).

References

1. Bagdanov, A., Worring, M.: Granulometric analysis of document images. In: IEEE International Conference on Pattern Recognition (ICPR), Québec, Canada, pp. 478–481 (August 2002)
2. Barnich, O., Jodogne, S., Van Droogenbroeck, M.: Robust analysis of silhouettes by morphological size distributions. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 734–745. Springer, Heidelberg (2006)
3. Barnich, O., Van Droogenbroeck, M.: Frontal-view gait recognition by intra- and inter-frame rectangle size distribution. Pattern Recognition Letters 30(10), 893–901 (2009)
4. Breen, E., Jones, R.: Attribute openings, thinnings, and granulometries. Computer Vision and Image Understanding 64(3), 377–389 (1996)
5. Piérard, S., Lejeune, A., Van Droogenbroeck, M.: 3D information is valuable for the detection of humans in video streams. In: Proceedings of 3D Stereo MEDIA, Liège, Belgium (December 2010)
6. Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Transactions on Image Processing 4(8), 1153–1160 (1995)
7. Soille, P.: Morphological image analysis: principles and applications. Springer, Heidelberg (1999)
8. Van Droogenbroeck, M.: Algorithms for openings of binary and label images with rectangular structuring elements. In: Talbot, H., Beare, R. (eds.) Mathematical Morphology, pp. 197–207. CSIRO Publishing, Sydney (2002)
9. Vincent, L.: Morphological area openings and closings for greyscale images. In: NATO Shape in Picture Workshop, pp. 197–208. Springer, Driebergen (1992)
10. Vincent, L.: Grayscale area openings and closings, their efficient implementation and applications. In: Mathematical Morphology and its Applications to Signal Processing, Barcelona, Spain, pp. 22–27 (May 1993)

Ultimate Opening and Gradual Transitions

Beatriz Marcotegui[1], Jorge Hernández[1], and Thomas Retornaz[2]

[1] MINES ParisTech, CMM- Centre de Morphologie Mathématique,
Mathématiques et Systèmes, 35 rue Saint Honoré - Fontainebleau, France
beatriz.marcotegui@mines-paristech.fr
http://cmm.ensmp.fr

[2] A2iA SA, Artificial Intelligence and Image Analysis
40 bis rue Fabert, 75007 Paris, France
thomas.retornaz@a2ia.com

Abstract. Ultimate Opening (UO) is a powerful operator based on numerical residues. In a multi-scale framework, it analyzes an image under a series of increasing openings. Contrasted objects are detected when they are filtered out by an opening, generating an important residue. Gradual transitions make this operator underestimate the contrast of blurred objects. In this paper we propose a solution to this problem, integrating series of non-null residues. The resulting operator handles correctly blurred boundaries, without modifying the behavior on sharp transitions.

Keywords: Numerical residues, ultimate opening, attribute opening, image segmentation, mathematical morphology.

1 Introduction

Given an image I, a residual operator ($r(I)$) in Mathematical Morphology is defined as the difference between two operators (Ψ and ζ) applied to the given image I: $r(I) = \Psi(I) - \zeta(I)$. Morphological gradient, defined as the residue of dilation and erosion ($\delta(I) - \epsilon(I)$), or top-hats, defined as the residue of the image and its opening ($I - \gamma(I)$) or its closing ($\phi(I) - I$) are residual operators extensively used. The extension of residues to a family of primitives Ψ_i and ζ_i is defined as the union of residues computed at different sizes i: $r(I) = \cup_i (\Psi_i(I) - \zeta_i(I))$. Ultimate erosion is an example of this type of operators. It is defined as the union of residues of erosions ($\Psi_i = \epsilon_i$) and openings by reconstruction ($\zeta_i = \gamma^{rec}$): $UltimateErosion(I) = \cup_i [\epsilon_i(I) - \gamma_i^{rec}(I)]$. Ultimate erosion is used for segmenting binary connected objects. Another example is the skeleton: the union of residues of erosions and the corresponding openings $Skeleton(I) = \cup_i [\epsilon_i(I) - \gamma_i(I)]$. The application of ultimate erosion or skeleton to binary images produces only one non-null residue (of index i) for each pixel. The union of these residues is then straightforward. This is not the case for gray level images. Thus, the extension of residual operators with a family of primitives to gray level images has to deal with the combination of several residues at different sizes for the same pixel. Beucher in [1] proposes to keep for

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 166–177, 2011.

each pixel the maximum residue and the index i corresponding to the size leading to it. Thus, he defines the quasi-distance function as the maximum residue between consecutive erosions ($QuasiDistance(I) = max_i\,(\epsilon_i(I) - \epsilon_{i+1}(I))$) and the ultimate opening (UO) as the maximum residue between consecutive openings ($UltimateOpening(I) = max\,(\gamma_i(I) - \gamma_{i+1}(I))$). Retornaz and Marcotegui in [2] introduce ultimate attribute openings, based on attribute operators [3], and Fabrizio [4] proposes an efficient implementation based on a max-tree representation. Leite and Guimarães propose an image filtering framework based on residues by attribute, but requires a set of parameters (size or complexity) in order to select which regions will be preserved [5].

(UO) is a powerful residual operator, able to segment generic images without a priori size information. It has been successfully used for several applications: granulometry of rocks [6], automatic text localization [2] and façade segmentation [7]. In this paper we focus on the behavior of this operator on gradual transitions and propose a solution to avoid underestimating the contrast of blurred objects. This paper is organized as follows: section 2 reminds the ultimate opening principle, section 3 analyzes the problem of gradual transitions and proposes a solution to deal with blurred objects, section A explains the implementation of the new operator, section 4 illustrates the performance of the new operator and finally, section 5 concludes the paper.

2 Ultimate Opening

2.1 Ultimate Opening Definition

Following a multi-scale approach, UO analyzes the image while performing a series of openings of increasing sizes: $\gamma_\lambda(I)$, with $\lambda = \{0, 1, ..., N-1\}$ and N the maximum opening size considered. The series of differences between consecutive openings, named residues ($r_\lambda(I) = \gamma_\lambda(I) - \gamma_{\lambda+1}(I)$), is computed. Each pixel keeps two significant pieces of information:

- the maximal residue, $R_\theta(I) = \max_\lambda(r_\lambda(I))$. It is the strongest change generated by an opening. An important structure is supposed to be filtered out by this opening and the corresponding residue estimates its contrast.
- the size of the opening leading to the maximal residue $q_\theta(I)$. If several openings lead to the same maximum $r_\lambda(I) = R_\theta(I)$, the largest λ among them is chosen (as proposed in the UO definition [1]). $q_\theta(I)$ is set at 0 for pixels where all residues are null. This happens for the minimum of the image or for objects larger than N, the largest opening size considered.

The definition of the UO is then written as:

Definition 1 (Ultimate opening [1]). *The ultimate opening operator, θ, of an image I is given by:*

$$\theta\,(I) : I \xrightarrow{\theta} (R_\theta\,(I)\,, q_\theta\,(I)) \tag{1}$$

where,

$$R_\theta\,(I) = \max_\lambda(r_\lambda(I)) = \max_\lambda\,(\gamma_\lambda(I) - \gamma_{\lambda+1}(I)) \tag{2}$$

$$q_\theta(I) = \begin{cases} max\,\{\lambda+1 \mid r_\lambda(I) = R_\theta(I)\} & R_\theta(I) > 0 \\ 0 & R_\theta(I) = 0 \end{cases} \quad (3)$$

where, $(\gamma_\lambda)_{\lambda\in\{0,1,\ldots N\}}$ *is an increasing family of openings.*

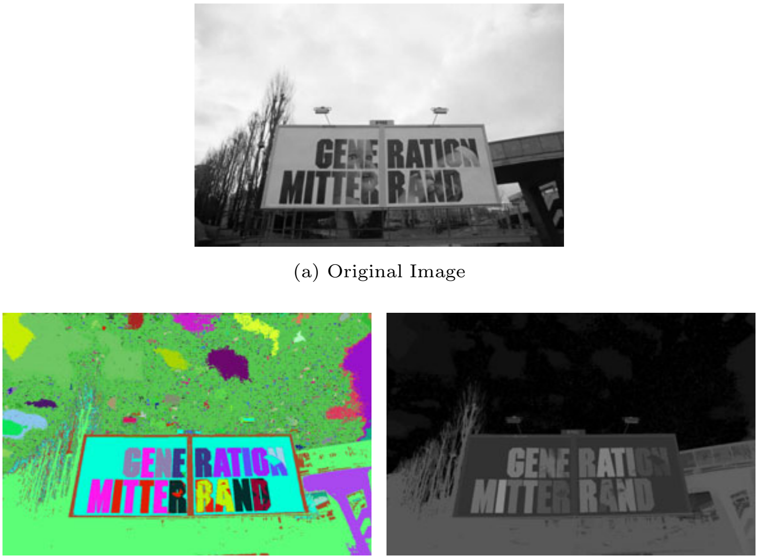
Labeling $q_\theta(I)$ we get a partial partition [8,9]: pixels are grouped in non intersecting segments that do not cover the whole domain ($q_\theta(I)$ is set at 0 for pixels where all residues are null).

Ultimate opening segments light objects in a dark background. In order to deal with dark objects in a light background, an ultimate closing should be used. The ultimate closing can also be applied to the gradient image, dealing with both polarities at the same time. The use of a gradient image is not recommended for images containing thin objects.

In this paper we focus on attribute openings. If not specified in the text, height attribute (y-extent of connected components) will be used in the following.

2.2 Example of Application

Fig. 1 shows the ultimate height closing of a real image 1(a). The letters of the image are not homogeneous. They are filled with a texture from another image. Fig. 1(b) shows $q_\theta(I)$ and Fig. 1(c) $R_\theta(I)$. In spite of the fact that the letters



(a) Original Image

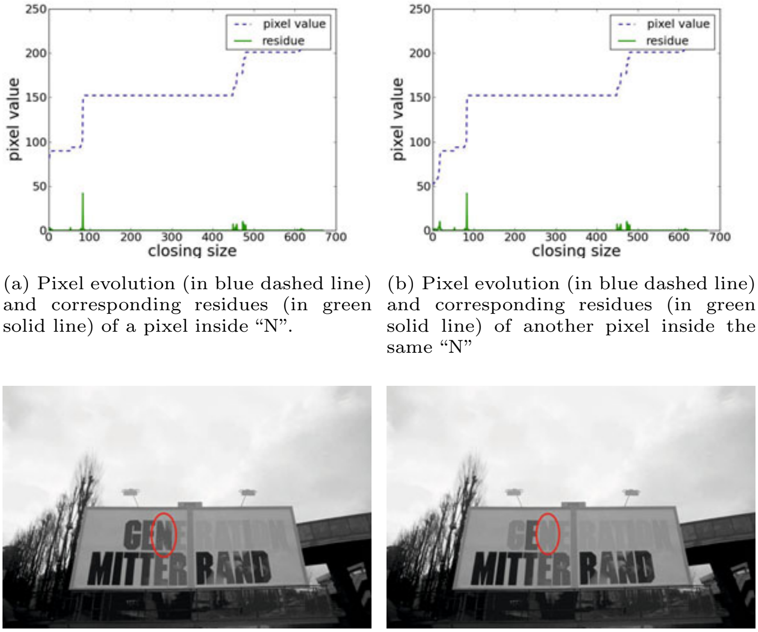
(b) $q_\theta(I)$ (size information)

(c) $R_\theta(I)$ (contrast information)

Fig. 1. Ultimate Height Closing : (b) $q_\theta(I)$, represented in false colors in order to see the segmented regions and (c) $R_\theta(I)$

are textured, most of them are correctly segmented, because their texture is less contrasted than the contrast between the characters and their background.

In order to understand this process, let us see the evolution with the series of closings (characters are darker than their background) of two pixels inside the same letter. Two pixels inside the first letter “N” of “GENERATION” are randomly chosen. The series of values of these two pixels, after applying closings of increasing sizes, are shown in blue dashed line in Fig. 2(a) and 2(b). The corresponding residues (the derivative of the blue dashed curve) are illustrated in green solid curves of Fig. 2(a) and Fig. 2(b). Residues for small closing sizes correspond to the internal fluctuations of the letter. These residues are different for different pixels of the same letter. However, both pixels have their highest residue for a closing of size 83. This is because the letter is 82 pixels high, and it is filtered out by a closing of size 83 (see Fig. 2(c) and Fig. 2(d)). The same important residue is seen by all the pixels of the structure. Thus, ultimate closing catches contrasted structures when they are filtered out, leading to interesting segmentation results without a priori information.



(a) Pixel evolution (in blue dashed line) and corresponding residues (in green solid line) of a pixel inside “N”.

(b) Pixel evolution (in blue dashed line) and corresponding residues (in green solid line) of another pixel inside the same “N”

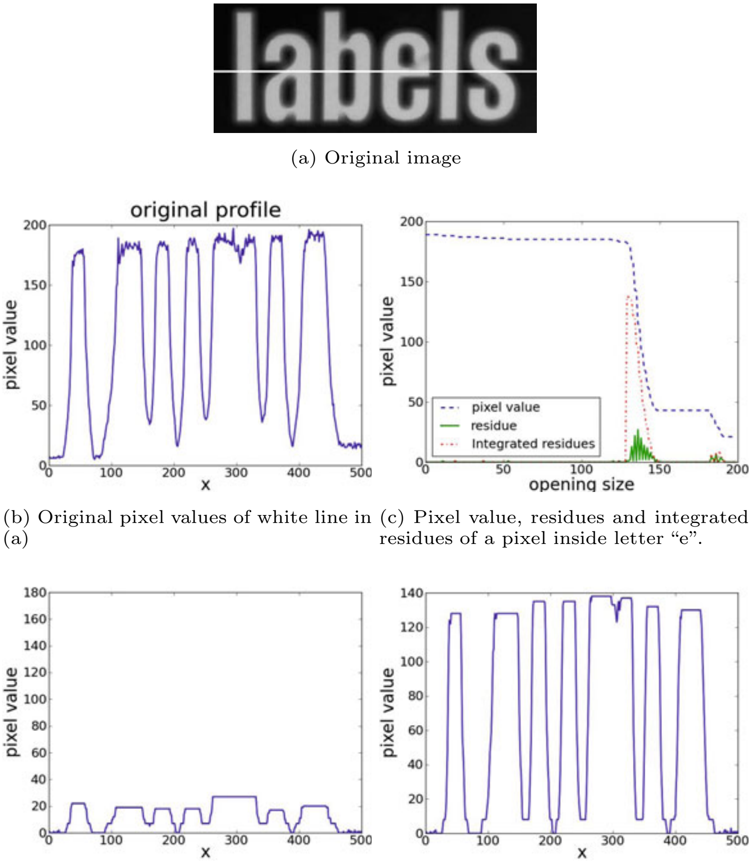
(c) Closing of size 82

(d) Closing of size 83

Fig. 2. Intermediate images of Ultimate Height Closing

3 Gradual Transitions and Δ Ultimate Opening

Ultimate opening produces interesting segmentation results in very diverse situations. Nevertheless, it underestimates the contrast of blurred objects. Indeed, the



(a) Original image

(b) Original pixel values of white line in (a)

(c) Pixel value, residues and integrated residues of a pixel inside letter "e".

(d) $R_\theta (I)$

(e) $R_\theta^\Delta (I)$

Fig. 3. Residues of a gradual transition (blurred objects)

boundary of a blurred object is a gradual transition. The contrast of the object is then divided into several steps. The contrast associated by UO to the object would be the largest of these steps, inevitably smaller than the real contrast of the object. An example of this situation is shown in Fig. 3. Fig. 3(a) shows the original image and Fig. 3(b) the profile of a horizontal line (superimposed in white in Fig. 3(a)). The gray level value of the letters is about 160 while the background value is about 10. The contrast of letters is then about 150. But the contrast estimated by the UO is only about 20 (see R_θ in Fig. 3(d)). Fig. 3(c) shows the pixel value evolution (in blue dashed line) of a pixel inside a letter, with a series of increasing openings. The corresponding residues are shown in green solid line. We can observe a series of non null residues for consecutive opening sizes, corresponding to the gradual transitions of the blurred boundary. In fact, transition regions are characterized by their small size. This is why small residues appear for consecutive opening sizes. If we integrate the series of non

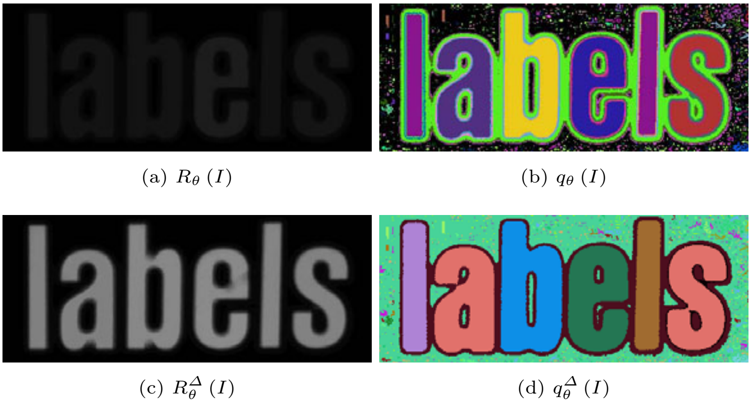

(a) $R_\theta(I)$
(b) $q_\theta(I)$
(c) $R_\theta^\Delta(I)$
(d) $q_\theta^\Delta(I)$

Fig. 4. UAO and ΔUAO comparison

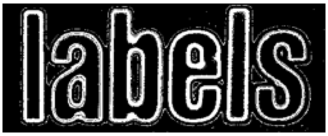


Fig. 5. Transition zones, detected as those that benefit from the residues integration

null residues, assuming that they correspond to transition zones, we get a much better contrast estimation. Red dashdotted line in Fig. 3(c) shows the integrated residues. Fig. 5 shows which pixels benefit from the integration. We can see that they are located in the boundary of the blurred objects.

This idea has been generalized to introduce the ΔUO, that integrates the series of residues until finding a series of Δ null residues. For $\Delta = 0$ we get the classic UO, $\Delta = 1$ integrates series of non-null residues, $\Delta = 2$ integrates series of residues separated by at least 2 consecutive null residues, and so on. The larger Δ is, the larger transition zones can be.

Fig. 4(a) and Fig. 4(b) show R_θ and q_θ of 3(a) while Fig. 4(c) and Fig. 4(d) show R_θ^Δ and q_θ^Δ with height attribute openings, and $\Delta = 1$. We can appreciate a much better contrast estimation of ΔUAO.

4 Results

In this section we report several results, illustrating the performance of ΔUAO with $\Delta = 1$ in several complex images from the public ICDAR 2003 image database [10]. In order to deal with both polarities ΔUAO is applied to image I and to its inverse I^c. For each pixel, the polarity leading to a bigger residue is kept (see Eq.4).

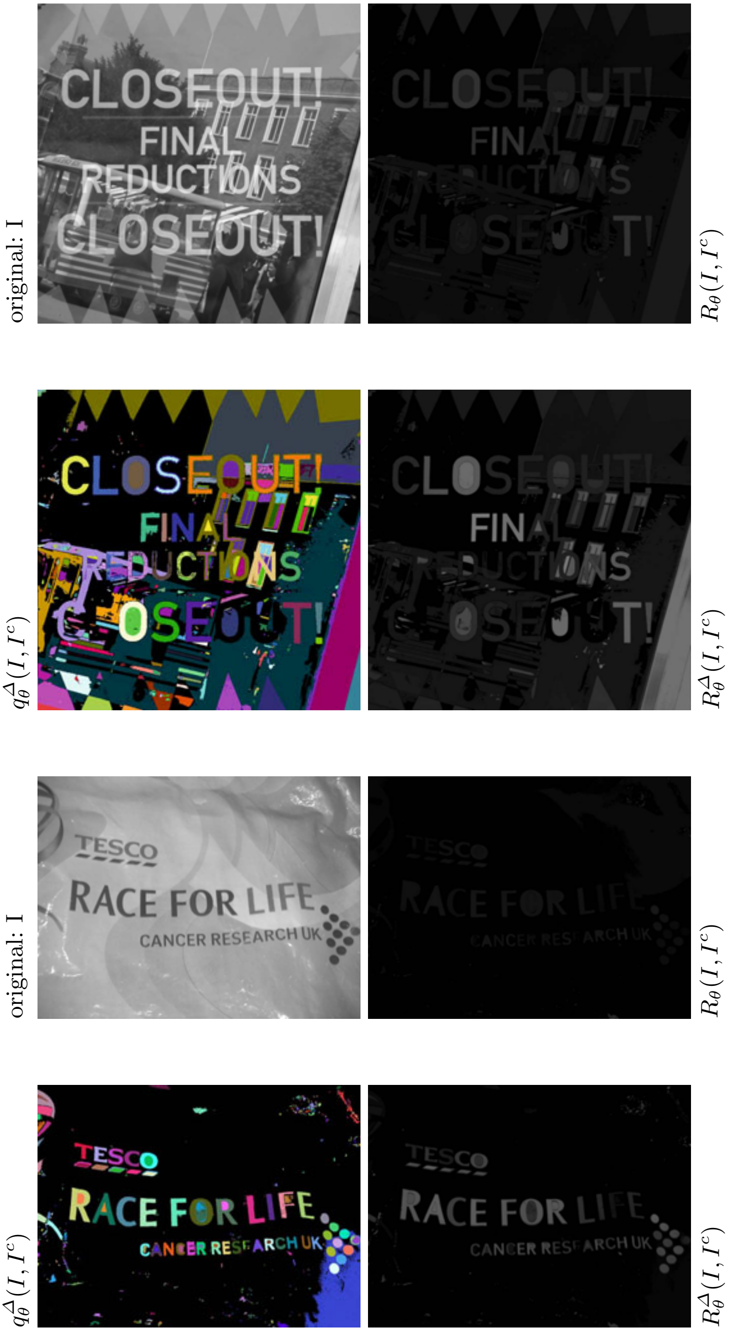


Fig. 6. *ΔUAO* Results of Δ ultimate height opening. For each image, first row: original image and R_θ; second row: $q_\theta^\Delta\,(I, I^c)$ and $R_\theta^\Delta\,(I, I^c)$.

$$\begin{aligned} R_\theta\left(I, I^c\right) &= \max\left(R_\theta\left(I\right), R_\theta\left(I^c\right)\right) \\ q_\theta\left(I, I^c\right) &= \begin{cases} q_\theta\left(I\right) & R_\theta\left(I\right) > R_\theta\left(I^c\right) \\ q_\theta\left(I^c\right) & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

The gradient is not used in order to be able to detect thin objects.

Fig. 6 compares classical UAO with ΔUAO. For each image, the first row shows the original image (on the left) and the classical UAO residue, according to equation 4. The contrast of the selected structures is clearly underestimated. In the second row, $q_\theta^\Delta(I, I^c)$ (on the left), represented in false colors in order to see the segmented regions and its corresponding residual information $R_\theta^\Delta(I, I^c)$ (on the right). Satisfactory results have been obtained in difficult situations: complex background or illumination problems. Most letters have been correctly segmented in q_θ^Δ and their associated contrast in R_θ^Δ corresponds to their real contrast.

5 Conclusions

In this paper we analyze the behavior of ultimate opening on blurred objects and see that it underestimates their contrast. We propose a solution, the Δ ultimate opening, that integrates series of residues, getting a much better contrast estimation. Moreover, ΔUO can be used as a detector of blurred contours, without any additional parameter. They are defined as the set of pixels that benefit from the integration.

We propose an efficient implementation of ΔUAO, as a simple modification from the max-tree based UAO implementation.

Interesting segmentation results for text segmentation in generic images are reported.

In the future, other criteria to combine residues can be studied. For example, the region stability, stablishing some links with Maximal Stable Extremal Regions [11]. We also intend to analyze the influence of the chosen attribute on UAO result.

References

1. Beucher, S.: Numerical residues. Image Vision Computing 25(4), 405–415 (2007)
2. Retornaz, T., Marcotegui, B.: Scene text localization based on the ultimate opening. In: International Symposium on Mathematical Morphology, vol. 1, pp. 177–188 (2007)
3. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. Computer Vision and Image Understanding 64(3), 377–389 (1996)
4. Fabrizio, J., Marcotegui, B.: Fast implementation of the ultimate opening. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 272–281. Springer, Heidelberg (2009)

5. Leite, N.J., Guimarães, S.J.F.: Morphological residues and a general framework for image filtering and segmentation. EURASIP J. Appl. Signal Process. 2001, 219–229 (2001)
6. Outal, S., Beucher, S.: Controlling the ultimate openings residues for a robust delineation of fragmented rocks. In: The 10th European Congress of Stereology and Image Analysis (2009)
7. Hernández, J., Marcotegui, B.: Ultimate attribute opening segmentation with shape information. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 205–214. Springer, Heidelberg (2009)
8. Serra, J.: A lattice approach to image segmentation. J. Math. Imaging Vis. 24, 83–130 (2006)
9. Ronse, C.: Partial partitions, partial connections and connective segmentation. J. Math. Imaging Vis. 32, 97–125 (2008)
10. ICDAR, Robust reading and locating database (2003), http://algoval.essex.ac.uk/icdar/TextLocating.html
11. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proc. of the British Machine Vision Conference, pp. 384–393 (2002)
12. Hernández, J.: Analyse morphologique d'images pour la modélisation d'environnements urbains, Ph.D. thesis, Mines ParisTech - C.M.M., Fontainebleau - France (December 2009)

A ΔUAO Implementation on Max-Tree

ΔUAO can be implemented easily and efficiently on a max-tree representation. As explained in [4], UAO can be computed in a single tree traversal, from the root to the leaves, with the following formula:

$$r_\lambda\left(node\right) = \begin{cases} t(node) - t(parent) + r_\lambda\left(parent\right) & \kappa(parent) == \kappa(node) \\ t(node) - t(parent) & \text{otherwise} \end{cases} \tag{5}$$

where $node$ and $parent$ are two linked nodes of the max-tree, $t(node)$ the gray level associated to $node$ and $\kappa(node)$ its corresponding attribute.

In order to implement ΔUAO, the condition $(\kappa(parent) == \kappa(node))$ is replaced by $(\kappa(parent) - \kappa(node)) \leq \Delta$ in the previous equation. The reason for that is the following: an opening of size $\kappa(node)+1$ will filter out the region corresponding to the $node$, assigning to it the value of its $parent$. Thus, the following opening producing a residue in this region would be the opening of size $\kappa(parent) + 1$, that removes the $parent$ node. Therefore, openings from size $\kappa(node) + 2$ to size $\kappa(parent)$ (both included) will not modify the region, leading to a series of $\kappa(parent) - \kappa(node) - 1$ null-residues. Given that, according to ΔUAO definition, residues should be integrated until finding a series of Δ null residues, the condition to this integration is set at $(\kappa(parent) - \kappa(node) \leq \Delta)$. The formula for r_λ^Δ computation becomes:

$$r_\lambda^\Delta(node) = \begin{cases} t(node) - t(parent) + r_\lambda^\Delta(parent) & \kappa(parent) - \kappa(node) \leq \Delta \\ t(node) - t(parent) & \text{otherwise} \end{cases} \tag{6}$$

The pseudo-code for ΔUAO is shown in algorithm 1. R_θ^Δ is computed from r_λ^Δ as explained in [4,12]. The process starts at the root node: $R_\theta^\Delta(root)$ and $q_\theta^\Delta(root)$ are initialized to zero. Then, function ComputeNode is called for each root child. ComputeNode function (see algorithm 2) computes $r_\lambda^\Delta(node)$ and compares it with $R_\theta^\Delta(parent)$. $R_\theta^\Delta(node)$ keeps the maximum value between them: $R_\theta^\Delta(node) = max(r_\lambda^\Delta(node), R_\theta^\Delta(parent))$. q_θ^Δ computation requires a particular attention. By definition it should be set at the size of the opening producing the highest residue. But several openings contribute to R_θ^Δ. Which size should be chosen for q_θ^Δ? If $R_\theta^\Delta(parent)$ is higher than $r_\lambda^\Delta(node)$, $R_\theta^\Delta(parent)$ and $q_\theta^\Delta(parent)$ are propagated to the *node* (lines 14 and 15 in algorithm 2). Otherwise, if $r_\lambda^\Delta(node)$ becomes $R_\theta^\Delta(node)$, $q_\theta^\Delta(node)$ is set at:

- $\kappa(node)+1$, if the *node* does not belong to an integration series ($is\Delta$ is false) or if *node* is the first node of the integration series when a maximal residue is produced ($is\Delta$ is true but *q_propagation* is false; lines 20 and 21).
- $q_\theta^\Delta(parent)$, until the end of the integration process ($is\Delta$ and *q_propagation* are both true; lines 18 and 19).

Thus, q_θ^Δ is set at the largest opening size involved in the integration process, when a maximum residue is produced. This size corresponds to the actual size of the detected region.

An example of r_λ^Δ computation, with $\Delta = 1$, for a synthetic profile of Fig. 7a is illustrated in Fig. 8. For the sake of simplicity, tree nodes are labelled with their gray level value (which is obviously not the case in the software implementation).

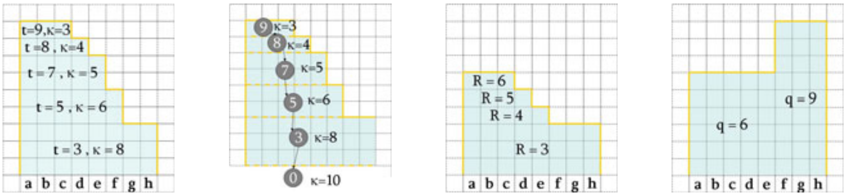
- The residue of the root node (0) is initialized to zero: $r_\lambda^\Delta(0) = 0$. $R_\theta^\Delta(0)$ and $q_\theta^\Delta(0)$ are also initialized to 0.
- Then, node 3 is processed: $r_\lambda^\Delta(3) = r_\lambda(3) = 3$. No integration is performed, since $\kappa(3) - \kappa(0) = 10 - 8 = 2 > \Delta$. $r_\lambda^\Delta(3) > R_\theta^\Delta(0)$, then $R_\theta^\Delta(3) = r_\lambda^\Delta(3)$ and $q_\theta^\Delta(3) = \kappa(3) + 1 = 9$.
- After that, node 5 is processed: $r_\lambda^\Delta(5) = r_\lambda(5) = 2$. No integration is performed, because $\kappa(5) - \kappa(3) = 8 - 6 = 2 > \Delta$. Since $r_\lambda^\Delta(5) < R_\theta^\Delta(3)$, R_θ^Δ and q_θ^Δ are propagated from the parent (node 3) to the child (node 5): $R_\theta^\Delta(5) = R_\theta^\Delta(3) = 3$ and $q_\theta^\Delta(5) = q_\theta^\Delta(3) = 9$.
- When computing $r_\lambda^\Delta(7)$, $r_\lambda(7)$ should be added to $r_\lambda^\Delta(5)$: $r_\lambda^\Delta(7) = r_\lambda(7) + r_\lambda^\Delta(5) = 2 + 2$, because $\kappa(7) - \kappa(5) = 6 - 5 = 1 \leq \Delta$. This residue is larger than $R_\theta^\Delta(parent)$, then $R_\theta^\Delta(7) = r_\lambda^\Delta(7)$ and $q_\theta^\Delta(7) = \kappa(7) + 1 = 6$. The variable *q_propagation* is activated for the rest of the integration series.
- This integration continues till node 9 (attributes have consecutive values: from 6 to 3), leading to $r_\lambda^\Delta(9) = 2 + 2 + 1 + 1 = 6$. Thus $R_\theta^\Delta(8) = r_\lambda^\Delta(8) = 5$ and $R_\theta^\Delta(9) = r_\lambda^\Delta(9) = 6$.

Algorithm 1. Compute Delta Ultimate Attribute Opening

```
Compute$\Delta UAO$()
  // Initialization
$R_\theta^\Delta(root) = 0$
$q_\theta^\Delta(root) = 0$
$q\_propagation = 0$
  // Compute children
**foreach** $child$ $in$ $Children(root)$ **do**
    ComputeNode($child$,$root$,$q\_propagation$)
```

Algorithm 2. Compute Node

```
ComputeNode($node$,$parent$,q_propagation)
  // Verify $\Delta$ attribute
$is\Delta = (\kappa(parent) - \kappa(node)) \leq \Delta$
  // Compute residue $r_\lambda^\Delta(node)$
**if** $(is\Delta)$ **then**
    $r_\lambda^\Delta(node) = t(node) - t(parent) + r_\lambda^\Delta(parent)$
**else**
    $r_\lambda(node)^\Delta = t(node) - t(parent)$
  // Compute $R_\theta^\Delta(node)$ and $q_\theta^\Delta(node)$
**if** $(R_\theta^\Delta(parent) > r_\lambda^\Delta(node))$ **then**
    $R_\theta^\Delta(node) = R_\theta^\Delta(parent)$
    $q_\theta^\Delta(node) = q_\theta^\Delta(parent)$
**else**
    $R_\theta^\Delta(node) = r_\lambda^\Delta(node)$
    **if** $(is\Delta \wedge q\_propagation)$ **then**
        $q_\theta^\Delta(node) = q_\theta^\Delta(parent)$
    **else**
        $q_\theta^\Delta(node) = \kappa(node) + 1$
  // Compute children
**foreach** $child$ $in$ $Children(node)$ **do**
    ComputeNode($child$,$node$,$is\Delta$)
```



(a) Profile

(b) Profile + *Max-Tree*

(c) R_θ^Δ

(d) q_θ^Δ

Fig. 7. **(a)** Profile with graylevels (t) and attribute (κ), **(b)** Max-Tree. **(c)-(d)** ΔUAO outputs

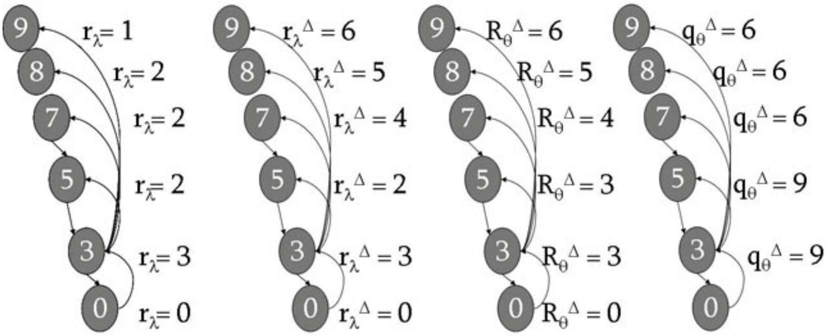


Fig. 8. ΔUAO on Max-tree

Spatio-temporal Quasi-Flat Zones for Morphological Video Segmentation

Jonathan Weber[1], Sébastien Lefèvre[2], and Pierre Gançarski[1]

[1] University of Strasbourg - LSIIT
{j.weber,gancarski}@unistra.fr
[2] University of South Brittany - Valoria
sebastien.lefevre@univ-ubs.fr

Abstract. In order to face the various needs of users, user-driven segmentation methods are expected to provide more relevant results than fully automatic approaches. Within Mathematical Morphology, several user-driven approaches have been proposed, mostly relying on the watershed transform. Nevertheless, Soille (IEEE TPAMI, 2008) has recently suggested another solution by gathering puzzle pieces computed as Quasi-Flat Zones (QFZ) of an image. In this paper, we study more deeply this user-driven segmentation scheme in the context of video data. Thus we also introduce the concept of Spatio-Temporal QFZ and propose several methods for extracting such zones from a video sequence.

Keywords: Quasi-flat zones, video segmentation, segmentation personalization.

1 Introduction

Following the increase of textual and then image data in personal databases and Web repositories, we are currently facing the same evolution with video data. Many video processing schemes or related use cases require a prior segmentation to get the objects-of-interest to be further processed. However, the segmentation of a given video is often not unique and depends on user needs. Thus it is necessary to rely on a segmentation method able to provide a personalized result.

Video segmentation methods designed within the framework of Mathematical Morphology may be clustered in two categories: automatic methods [2,4] which do not require any user interaction (apart from parameter settings) and interactive methods [6,7,8] (but also video extension of [10]) where the user has mainly to draw some markers over objects-of-interest in order to drive the segmentation process. Results returned by automatic methods are then not adapted to user needs and often face the problem of over-segmentation. Interactive methods are more time-consuming (from a user point of view) but provide a personalized result. Another solution to solve the problem of segmentation personalization is to provide an over-segmentation which will then be reduced by the user through region merging in order him to obtain the expected segmentation. Image over-segmentation may be achieved using flat zones [11] but it then results in an

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 178–189, 2011.

extreme over-segmentation. Quasi-Flat Zones have been introduced in order to reduce this over-segmentation while keeping interesting properties of flat zones. QFZ are based on a less restrictive criteria to build the regions, thus leading to larger regions, while keeping a low computational cost and region borders able to represent most of the frontiers between the objects-of-interest. Besides, Soille [13] notices that QFZ are not really segmentation methods but rather methods which split an image into puzzle pieces. Identifying the QFZ is then a preprocessing step in an image segmentation process based on merging of puzzle pieces. This merging may be driven by the user, thus solving the problem of segmentation personalization. Let us observe however that there is no definition of QFZ for video sequences yet.

In this article, we recall the QFZ definition in the framework of logical predicate connectivity introduced by Soille [12,15]. We then extend this definition to video sequences, study how it can be applied to video segmentation personalization and present some preliminary results to show its interest. Finally we give some conclusions and indicate future research directions.

2 QFZ-Based Image Segmentation

2.1 Logical Predicate Connectivity

QFZ rely on the concept of α-connected paths. A path is said α-connected if all paths between any pair of its pixels are Lipschitz-continuous, thus leading to the following definition.

Given a neighbourhood N, a path $\mathcal{P}$ made of n pixels $(p_0, p_1, ..., p_{n-1})$ is an α-connected path (α-$\mathcal{P}$) if and only if:

$$\forall i \in [0, n-2], p_i \in N(p_{i+1}) \text{ and } |f(p_i) - f(p_{i+1})| \leq \alpha \tag{1}$$

We note α-$\mathcal{P}(p, q)$ the set of α-connected paths between p and q. This notion let us define the most simple QFZ, i.e., α-connected zones [9] which will be noted here α-CC. An α-CC is defined as:

$$\alpha\text{-}CC(p) = \{p\} \cup \{Q \mid \forall q \in Q, \alpha\text{-}\mathcal{P}(p, q) \neq \emptyset\} \tag{2}$$

The α-CC of a pixel p is then the set of pixels to which it is linked through an α-connected path. Let us observe that flat zones are a particular case of α-CC with $\alpha = 0$. The α-CC have the following hierarchical property which will be useful later in this paper:

$$\forall \alpha' \leq \alpha, \alpha'\text{-}CC(p) \subseteq \alpha\text{-}CC(p) \tag{3}$$

Segmenting an image into α-CC can result in an under-segmentation. If α is set too high, it will lead to a chaining effect, which may even result on a single QFZ for the whole image (it depends of course of the image under consideration and the selected α value). In order to counter this problem, new QFZ definitions based on α-CC have been elaborated. In a goal of unification of existing works, Soille and Grazzini [12,15] have proposed a theoretical framework called logical predicate connectivity (we recall that a logical predicate P returns true when the parameter satisfies the predicate, false otherwise). They define a new kind of QFZ (noted $(P_1, ..., P_n)$-CC here) which leads to QFZ satisfying all the n logical predicates. Various predicates may be involved, for instance: global range predicate which checks if the difference between minimal and maximal values of pixels within a QFZ is below a threshold (ω); connectivity index which is the ratio between the number of 2-pixels α-connected paths and the number of 2-pixels paths within a QFZ. This predicate is verified if the index is higher than a threshold (β). The $(P_1, ..., P_n)$-CC thus consists in seeking, for each pixel p, the largest α-CC satisfying all the predicates. Thanks to the property 3, we know that if $\alpha' < \alpha$ then α'-$CC(p)$ is a subset or equal to α-$CC(p)$. When predicates are not verified for a given value of α, we can use this property to decide to decrement α in order to check if the predicates are verified for a lower value and to loop until finding the maximal value of α for which all the predicates are verified:

$$(P_1, ..., P_n)\text{-}CC(p) = \bigvee \left\{ \alpha'\text{-}CC(p) \;\middle|\; \begin{array}{r} \forall k \in \{1, ..., n\},\ P_k\left(\alpha'\text{-}CC(p)\right) = true \\ \forall \alpha\text{"} \leq \alpha', \forall q \in \alpha'\text{-}CC(p), \quad P_k\left(\alpha\text{"-}CC(q)\right) = true \end{array} \right\} \quad (4)$$

This theoretical framework is adapted to methods ensuring the unicity property. Indeed we are seeking the largest α'-CC verifying all logical predicates. It is thus not possible to consider methods which do not provide a unique QFZ segmentation. More than only a framework to unify existing definitions, the $(P_1, ..., P_n)$-CC also allows to elaborate new QFZ definitions. Three predicates are currently used within the QFZ: local range (α), global range (ω) and connectivity index (β). In the framework introduced by Soille and Grazzini, it is possible to include predicates related to other features (perimeter, area, etc.) but also to more complex descriptors (texture, gradient, etc.) as long as these predicates fulfill the condition defined in Eq. (4).

Some clues to using QFZ in multivariate images have been given by Soille [13]: α is assumed to be a vector with the same value in all components. Then, α may be easily ordered through a total ordering (decrementing $\alpha = (3, 3, 3)$ gives $\alpha = (2, 2, 2)$). Global range predicate is processed similarly, and is true only if it is verified marginally for all bands.

In the sequel of this article, we will denote by QFZ the colour QFZ built using $(P_1, ..., P_n)$-CC with only the global range predicate and a given α.

2.2 Filtering

QFZ suffer from the transition region problem. Transition regions are regions between two objects where a staircase phenomenon occurs on edge pixel values. This is due to the image discretization process and the subsequent value interpolation. This artefact leads to an over-segmentation near to the edges which will then be made of tiny QFZ. Some solutions have been proposed to solve this problem. Soille and Grazzini [15] define transition regions as QFZ containing only transition pixels. Every pixel which is not a local extremum is considered as a transition pixel. All QFZ corresponding to transition regions are removed, and remaining QFZ are enlarged using a region growing algorithm [1]. After the removal of these regions, the amount of flat zones is reduced significantly. The solution proposed by Soille and Grazzini does not depend of any parameter and relies on a precise definition of a transition region. But from our experiments, we have observed that many regions of a few pixels remain after applying their strategy. These regions do not fit with the definition of transition regions, but are still sources of a high over-segmentation. Thus, a more efficient and robust filtering method is still lacking.

Other authors have proposed QFZ filtering methods using a QFZ minimal area thresholding step. Angulo and Serra [3] suggest to merge QFZ characterized by an area lower than a given threshold with the most similar neighbouring QFZ. With this method, no more transition region is present in the final segmentation. Zanoguera [16] removes QFZ with an area below a given threshould (thus including transition regions) before applying a Watershed transform to enlarge remaining QFZ in areas where small QFZ have been removed. Soille [14] proposes a filtering method based on an iterative increase of the minimal area, followed at each iteration by both a region growing algorithm relying on QFZ with area greater or equal to minimal area and an image simplification algorithm. The simplified image will then be segmented into QFZ at the next iteration. This process is repeated until filtered QFZ become stable. Crespo *et al.* [5] deal with flat zones and propose a flat zones merging procedure. It consists in selecting significant flat zones (according to different criteria) and expand them by incorporating most similar adjacent non significant flat zones. Following some ideas introduced in these methods, we have designed another filtering method. It relies on the Seeded Region Growing (SRG) algorithm [1] but we apply it on the QFZ rather than on the pixels (cf. fig 1). To do so, we consider a minimal area threshold similarly to existing approaches. We set all QFZ with an area greater or equal to this threshold as seeds for the SRG algorithm which is applied on the region adjacency graph. We thus obtain a much more reduced over-segmentation compared to the result obtained without filtering. The highest the area threshold is, the more reduced the over-segmentation is. But in the same time, it is much more probable to obtain an under-segmentation of some objects-of-interest. Our region growing being applied on QFZ rather than on pixels, the proposed method requires only a low computational cost.

3 Extension to Video Data

3.1 Limits of a 3D Straight Extension

The most direct extension of QFZ to video sequences is to consider a video sequence as a 3-D spatio-temporal cube. We can reuse the existing definitions, thus changing only the neighbourhood considered (spatio-temporal rather than purely spatial).

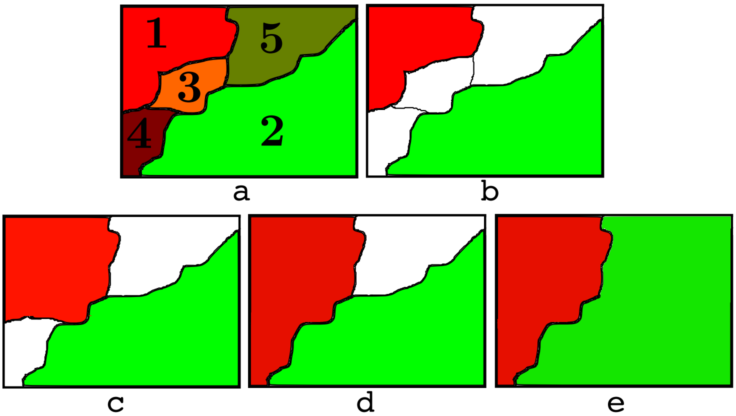


Fig. 1. Our QFZ filtering method: a) original QFZ, b) removal of QFZ with area lower than a given threshold, c) iteration #1 of SRG: QFZ 1 grows by incorporating pixels from QFZ 3, d) iteration #2 of SRG: QFZ 1 grows by incorporating pixels from QFZ 4, e) iteration #3 (last) of SRG: QFZ 2 grows by incorporating pixels from QFZ 5

Computing the $(P_1, ..., P_n)$-CC in $3D$, we obtain a higher spatial oversegmentation than in $2D$. On the 15 frames of *carphone* sample (Fig. 3.a) for $\alpha = \omega = 20$ (we will use these values in the sequel), we obtain on average $4,441CC$ per frame in $2D$ vs. $6,779CC$ per frame in $3D$ ($55,040CC$ on the full sample). Indeed, by analysing the video sequences in $3D$, the considered neighborhood contains more pixels and therefore an α-CC contains more pixels (see *chaining effect* discussed above). This naturally increases the risk of violating one of the considered predicates. Thus, the largest α-CC satisfying all predicates is often produced with a small α value. This leads to tiny QFZ of only a few pixels, while such QFZ are unusable for video segmentation.

3.2 Sequential Processing of Spatial and Temporal Dimensions

As $3D$ approach is not suitable for video processing, we consider rather the $2D+t$ approach. In this approach, we successively (and no more jointly) consider the spatial and temporal dimensions, as illustrated in Fig. 2. We discuss here first the spatial to temporal $(2D + t)$ approach and then the temporal to spatial $(t + 2D)$ approach.

With the spatial to temporal approach, QFZ are first built on each frame independently. Then they are considered as nodes of a graph which are valued (here we consider the QFZ mean value). Edges are then introduced to temporally connect QFZ from successive frames and overlapping spatial coordinates. Each edge is valued by the difference between related node values. The new QFZ are the largest connected components of nodes whose connecting edges have a value less or equal to α and which do not violate any predicate. We observe that $(P_1, ..., P_n)$-CC produced significantly fewer regions in $2D+t$ ($23,926CC$), thus reducing the extreme segmentation we noted in $3D$ ($55,040CC$). This can be

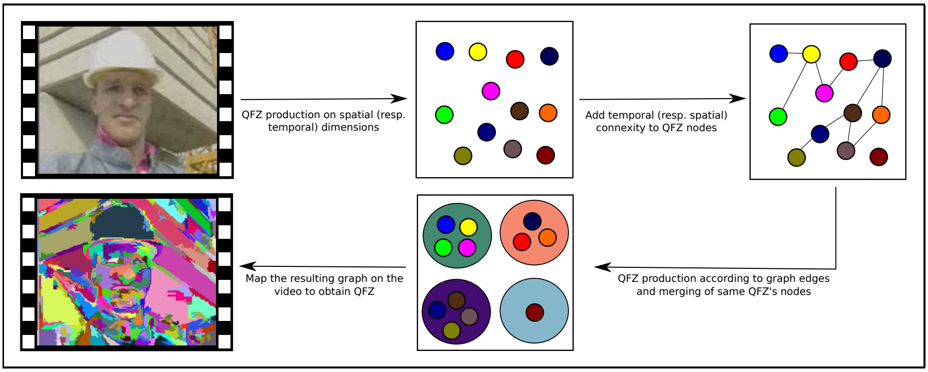

QFZ production on spatial (resp. temporal) dimensions
Add temporal (resp. spatial) connexity to QFZ nodes
QFZ production according to graph edges and merging of same QFZ's nodes
Map the resulting graph on the video to obtain QFZ

Fig. 2. Video Quasi-Flat Zones production by separated processing of spatial and temporal dimensions

explained by the distinct computing of the two dimensions (spatial and temporal). In $2D + t$ the first processing (spatial only) produces spatially wider QFZ reducing the spatial over-segmentation. But, then the second processing (temporal) introduces temporal over-segmentation. This is due to the predicate constraints: the regions being spatially more extensive, they are less homogeneous and therefore may have significantly different mean values which will violate a predicate during the temporal computing of $(P_1, ..., P_n)$-CC.

With the temporal to spatial approach, QFZ are first built for each spatial coordinate independently, according to the temporal dimension. After this temporal processing, we therefore obtained an extreme spatial over-segmentation since, for each frame, each pixel belongs to a different QFZ. Similarly to the $2D+t$ case, we consider QFZ as nodes of a graph and apply the same process as previously, but considering here the spatial dimension instead of the temporal one. We note that $(P_1, ..., P_n)$-CC produces fewer regions $(16, 830CC)$ than the $2D + t$ approach due to a smaller temporal over-segmentation.

Let us observe that the $(P_1, \ldots, P_n)$-CC highlights an interesting phenomenon. Due to their different order when processing spatial and temporal dimensions, the approaches $2D+t$ and $t+2D$ induce different over-segmentations: a reduced spatial but high temporal one for the former, and a higher spatial but reduced temporal one for the latter. Nevertheless, both approaches provide better results than the $3D$ approach. Selection between $2D + t$ and $t + 2D$ depends on the video under consideration. It may seem better to use the first approach with short videos of high-resolution, and to use the second for long videos of lower-resolution. Finally, let us note that the spatial and the temporal processing are both relying on $(P_1, ..., P_n)$-CC, which guarantees the uniqueness of the result. Thus they also ensure this fundamental property.

3.3 Filtering

The filtering methods presented in section 2.2 can be extended to video data. As far as our method based on a minimum area threshold is concerned, adaptation depends on the chosen approach. For the $3D$ approach, we could trivially extend the method and no longer consider a minimum area but a minimum volume. However, if considering a minimum volume would be effective in the context of truly three-dimensional images, it is not suitable for videos that are spatio-temporal and not purely spatial. Indeed, assuming a minimum volume, a QFZ having few pixels in a spatial area, but on many frames, would be kept despite the fact it is probably not an object but a part of an object. Thus, we use a threshold of minimum mean area, the mean area being computed as follows:

$$A_{mean} = \frac{\#\ \text{QFZ pixels}}{\#\ \text{frames where the QFZ is present}} \tag{5}$$

We filter the QFZ from approaches $2D + t$ and $t + 2D$ in the same manner as for the $3D$ approach, once the QFZ building is achieved.

Similarly to the image filtering, the oversegmentation is here strongly reduced. Indeed, setting the threshold of minimum mean area to 10 pixels, we get $686CC$ for $3D$, $845CC$ for $2D + t$ and $378CC$ for $t + 2D$. In addition, we get few or no under-segmentation, which makes obtained QFZ relevant for segmentation.

Filtering by minimum area threshold is very effective in reducing QFZ over-segmentation in video sequences. We obtain a very substantial over-segmentation reduction while maintaining the QFZ quality. By combining the definitions of video QFZ and the filtering area, we obtain an effective method of video pre-segmentation. This pre-segmentation can be used by QFZ merging methods to obtain user-personnalized segmentation.

4 User-Driven Video Segmentation

Encountered in image segmentation, over-segmentation is even more present when dealing with video segmentation. For instance, when processing the sample *carphone* (Fig. 3.a) with the Predictive Watershed [4], we obtained about 2000 regions. This problem happens obviously also with segmentation by quasi-flat zones: segmenting the same sequence by $(P_1, ..., P_n)$-CC $t{+}2D$, with $\alpha = \omega = 20$ and a minimal area of 10 also provides an over-segmentation ($378CC$). Moreover, the resulting segmentation is not personnalized. This drawback may be solved by relying on user interaction. Such an interaction aims the user to both customize the segmentation and reduce over-segmentation. User-driven segmentation is a well-known principle in Mathematical Morphology. The main interactive morphological tool for segmentation is the marker-based watershed [10]. It has been developed for image segmentation but can be easily extended to video data. There exists also some morphological methods dedicated to video data, such as the method proposed by Gu and Lee [7] which relies on an initial interactive segmentation of a frame by marker-based watershed. The propagation of this segmentation to following frames is performed using motion estimation and region tracking. It can be corrected by user interaction. Marcotegui *et al.* [8] propose a segmentation tool based on an initial interactive frame segmentation. This interactive segmentation is based on a multi-scale segmentation which is adapted to user needs by different tools: scale selection, region correction, region merging and marker drawing. The segmentation is propagated to the next frames via a partition projection which allows user interaction. The watershed from propagated markers method [6] is a more recent example. The first frame is segmented using marker-based watershed. Markers are extracted from the resulting segmentation and are propagated to the next frame by a motion estimation method. The next frame is segmented by marker-based watershed using the propagated markers. The user can modify the markers. In the context of QFZ, as indicated by Soille [13], a user-driven segmentation may correspond to the assembling of puzzle pieces.

We suggest a new principle for user-interactivity in video segmentation by defining a QFZ segmentation guided by markers. First, a base QFZ segmentation is produced. Then, the user draws markers on the video data. Thus, he customizes the segmentation by indicating his objects-of-interest. QFZ beneath the markers are considered as the seeds of a Seeded Region Growing algorithm [1]. The region growing will then merge the different QFZ according to their distance in terms of color, which can be related to an α parameter in the QFZ context. Since the user only see original video (and not QFZ), it is possible that several markers are found over the same QFZ. In this case, we consider that there are two possibilities: either the marker has been ill-drawed or the QFZ is ill-segmented. Here, we assume that the user has well-drawn the marker and that the QFZ has to be corrected. To do so, ill-segmented QFZ are partitioned using Seeded Region Growing with the user's markers as seeds. Thus, ill-segmented QFZ are corrected: it both improves the accuracy of the initial over-segmentation and solve the problem of having multiple markers over the same QFZ.

In order to evaluate the relevance of our proposal, we conducted some experiments on the *carphone* sequence. We compared the $(P_1, ..., P_n)$-CC $2D + t$ and $t + 2D$ methods to the Marker-Based Watershed known as the standard interactive segmentation method of Mathematical Morphology. We also compared interactive QFZ to the Seeded Region Growing involved in our method, in order to show how our method benefits from such algorithm and what it offers compared to a direct processing of Seeded Region Growing. In this perspective, we used three different settings of markers. First set of markers is just few points drawn on the median frame (Fig. 3.b). The second set is composed of heavy markers only drawn on the median frame (Fig. 3.c). The third and last set contains heavy markers on frame 4, 8 and 11 (Fig. 3.d-f). The results of these experiments are presented in Tab. 1. We denote here by precision the ratio of well-segmented pixels (affected to the good region). We used four sets of parameters (α, ω) for $(P_1, ..., P_n)$-CC to evaluate the impact of parameter settings (which have besides not been optimized). Let us observe that, whatever the markers used, our method always provides better results than the SRG. This shows that our method by applying SRG on QFZ is better than a direct computation of SRG on pixels. Concerning the comparison to marker-based watershed, the marker-based $(P_1, ..., P_n)$-CC is able achieve better results on each set of markers but only with appropriate parameter settings. Moreover, there is

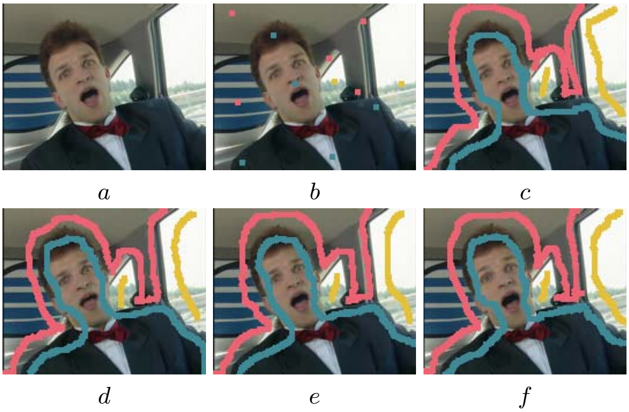


Fig. 3. Sets of markers on sequence *carphone* (a) frame 8 of extract from *carphone* sequence, b) few points on frame 8, c) heavy markers on frame 8, def) heavy markers on frame 4,8 and 11)

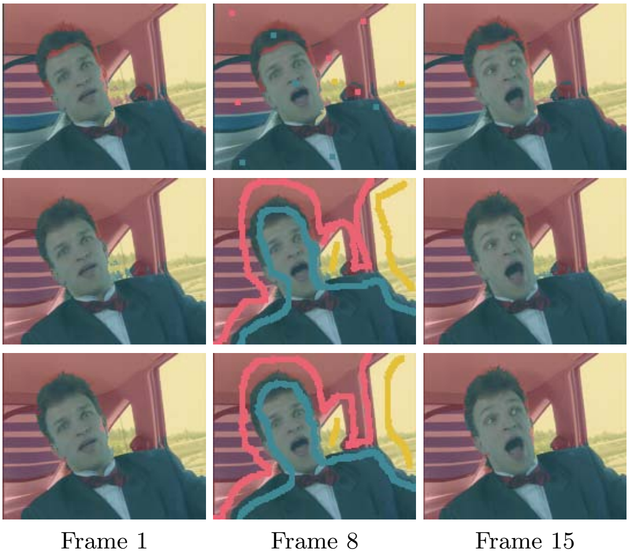


Fig. 4. Results of marker-based $(P_1, ..., P_n)$-CC $2D{+}t$ ($\alpha = \omega = 40$,$A_{mean} = 10$) using different set of markers, (top) points on frame 8, (middle) heavy markers on frame 8, (bottom) heavy markers on frame 4,8 and 11

no single parameter combination which achieves better results with all sets of markers. But, if we exclude the first set of markers (few points in the middle frame) which are clearly insufficient, we observe that $(P_1, ..., P_n)$-CC $t{+}2D$ and $\alpha = \omega = 30$ produces the best results with both remaining sets of markers. However, like the other two interactive methods, the interactive QFZ segmentation is very sensitive to the markers given by the user. This is illustrated by Fig. 4

Table 1. Comparison of pixel segmentation precision with different markers (a) a few points on the median frame, b) heavy markers one the median frame, c) heavy markers on three frames)

Method	**Parameters**	**Precision**		
		(a)	**(b)**	**(c)**
$(P_1,\ldots,P_n)$-CC $2D+t$	$\alpha=\omega=10$	0.837	0.982	0.981
$(P_1,\ldots,P_n)$-CC $2D+t$	$\alpha=\omega=20$	0.841	*0.987*	0.989
$(P_1,\ldots,P_n)$-CC $2D+t$	$\alpha=\omega=30$	*0.851*	0.981	0.989
$(P_1,\ldots,P_n)$-CC $2D+t$	$\alpha=\omega=40$	*0.882*	*0.987*	0.989
$(P_1,\ldots,P_n)$-CC $t+2D$	$\alpha=\omega=10$	**0.899**	0.968	0.970
$(P_1,\ldots,P_n)$-CC $t+2D$	$\alpha=\omega=20$	0.828	0.979	0.984
$(P_1,\ldots,P_n)$-CC $t+2D$	$\alpha=\omega=30$	0.814	**0.988**	**0.993**
$(P_1,\ldots,P_n)$-CC $t+2D$	$\alpha=\omega=40$	0.837	*0.986*	0.988
Marker-Based Watershed		0.851	0.985	0.990
Seeded Region Growing		0.802	0.806	0.824

Table 2. Comparison of pixel segmentation precision between $(P_1,...,P_n)$-CC and watershed from propagated markers

Method	**Parameters**	**Precision**
$(P_1,\ldots,P_n)$-CC $2D+t$	$\alpha=\omega=10$	*0.985*
$(P_1,\ldots,P_n)$-CC $2D+t$	$\alpha=\omega=20$	*0.984*
$(P_1,\ldots,P_n)$-CC $2D+t$	$\alpha=\omega=30$	0.978
$(P_1,\ldots,P_n)$-CC $2D+t$	$\alpha=\omega=40$	0.979
$(P_1,\ldots,P_n)$-CC $t+2D$	$\alpha=\omega=10$	0.964
$(P_1,\ldots,P_n)$-CC $t+2D$	$\alpha=\omega=20$	0.966
$(P_1,\ldots,P_n)$-CC $t+2D$	$\alpha=\omega=30$	*0.983*
$(P_1,\ldots,P_n)$-CC $t+2D$	$\alpha=\omega=40$	**0.988**
Watershed From Propagated Markers		0.983

which shows the results obtained by marker-based $(P_1,...,P_n)$-CC $2D+t$ with parameters $\alpha=\omega=40$ and $A_{mean}=10$ with the different sets of markers.

We also compared our method to a more recent method, the Watershed from Propagated Markers [6], for which we used the binding of markers and the region-based motion propagation. As the objective here was to compare our method to a recent interactive approach in similar conditions (here the time required for the user), we did not allow the marker correction by the user and marked only the first frame. The results are given in Tab. 2 and show that marker-based $(P_1,...,P_n)$-CC in these conditions can be more accurate than the Watershed from Propagated Markers depending on the parameter settings. It implies that even if marker-based $(P_1,...,P_n)$-CC do not use motion information, its computing by assembling spatio-temporal puzzle pieces allows to obtain results comparable to other methods using motion estimation (e.g., [6]).

5 Conclusion

In this paper, we proposed both an extension of QFZ to video sequences and an interactive method for assembling these QFZ in order to build a user-personnalized segmentation. The separate processing of spatial and temporal dimensions improves the segmentation compared to a straight three-dimensional processing of video data. The proposed method for assembling QFZ according to the user's needs is intuitive and provides good results compared to other existing methods.

Our future work will focus on a better validation of our method. We have presented preliminary results in this paper, but the method needs a deeper evaluation on a larger dataset to assess its validity. We will also focus on improving the markers. Indeed, the video is currently only marked before the processing. However, it seems relevant to be able to correct markers (like what is done in [6,7,8]) in order to iteratively improve the segmentation. Indeed the eventual correction of some QFZ will improve the over-segmentation at each iteration. Moreover, most of the computational cost of our approach is linked to the initial QFZ segmentation: the marker-based QFZ merging is very efficient because it is performed on the QFZ adjacency graph. We also consider to apply video QFZ on other data spaces, such as optical flow values instead of pixel values. Moreover, we plan to improve the QFZ merging process by using other features than only the color mean value. Finally, a comparative study must be achieved to determine optimal parameter settings (α, ω and A_{mean}) able to produce good results on a wide variety of video sequences.

Finally, as our method is based on a graph reduction process, we would like to design a machine learning scheme to understand how to perform segmentation from this reduction process. The idea is here to perform first a learning of some videos marked by the user and then to enable the system to segment unmarked, but simply over-segmented with QFZ, video sequences.

Acknowledgements

This work has been supported by Ready Business System, Entzheim, France and the French National Association for Research and Technology (ANRT). We particulary thank Christian Dhinaut from RBS for his support.

References

1. Adams, R., Bischof, L.: Seeded region growing. IEEE Transanction on Pattern Analysis and Machine Intelligence 16(6), 641–647 (1994)
2. Agnus, V.: Segmentation spatio-temporelle de séquences d'images par des opérateurs de morphologie mathématique. Ph.D. thesis, Université Louis Pasteur, Strasbourg (2001)
3. Angulo, J., Serra, J.: Color segmentation by ordered mergings. In: Proceedings of the IEEE International Conference on Image Processing, pp. 125–128 (2003)

4. Chien, S.-Y., Huang, Y.-W., Chen, L.-G.: Predictive watershed: a fast watershed algorithm for video segmentation. IEEE Transactions on Circuits and Systems for Video Technology 13(5), 453–461 (2003)
5. Crespo, J., Schafer, R.: The flat zone approach and color images. In: Serra, J., Soille, P. (eds.) Mathematical Morphology and its Applications to Image Processing, pp. 85–92. Kluwer Academic Publishers, Dordrecht (1994)
6. Flores, F.C., Lotufo, R.A.: Watershed from propagated markers: An interactive method to morphological object segmentation in image sequences. Image and Vision Computing 28(11), 1491–1514 (2010)
7. Gu, C., Lee, M.-C.: Semiautomatic segmentation and tracking of semantic video objects. IEEE Transactions on Circuits and Systems for Video Technology 8(5), 572–584 (1998)
8. Marcotegui, B., Zanoguera, F., Correia, P., Rosa, R., Mech, R., Wollborn, M.: A video object generation tool allowing friendly user interaction. In: IEEE International Conference on Image Processing, vol. 2, pp. 391–395 (1999)
9. Nagao, M., Matsuyama, T., Ikeda, Y.: Region extraction and shape analysis in aerial photographs. Computer Graphics and Image Processing 10(3), 195–223 (1979)
10. Rivest, J.-F., Beucher, S., Delhomme, J.: Marker-controlled segmentation: an application to electrical borehole imaging. Journal of Electronic Imaging 1(2), 136–142 (1992)
11. Serra, J., Salembier, P.: Connected operators and pyramids. In: Proceedings of SPIE, Non-Linear Algebra and Morphological Image Processing, vol. 2030, pp. 65–76 (1993)
12. Soille, P.: On genuine connectivity relations based on logical predicates. In: Proceedings of the 14th International Conference on Image Analysis and Processing, pp. 487–492. IEEE Computer Society, Washington, DC, USA (2007)
13. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(7), 1132–1145 (2008)
14. Soille, P.: Constrained connectivity for the processing of very-high-resolution satellite images. International Journal of Remote Sensing 31(22), 5879–5893 (2010), doi:10.1080/01431161.2010.512622
15. Soille, P., Grazzini, J.: Constrained connectivity and transition regions. In: Proceedings of the 9th International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing, Groningen, The Netherlands, pp. 59–69. Springer, Heidelberg (2009)
16. Zanoguera, F.: Segmentation interactive d'images fixes et de séquences vidéo basée sur des hierarchies de partitions. Ph.D. thesis, Ecole des Mines de Paris (2001)

Primitive and Grain Estimation Using Flexible Magnification for a Morphological Texture Model

Lei Yang[1], Liang Li[1], Chie Muraki Asano[2], and Akira Asano[1]

[1] Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima,
Hiroshima 739-8521, Japan
[2] Department of Lifestyle Design, Yasuda Women's University, Hiroshima,
Hiroshima 731-0153, Japan
{leiyang10,liliang1027}@hiroshima-u.ac.jp,
asano@yasuda-u.ac.jp,
asano@mis.hiroshima-u.ac.jp

Abstract. An improved morphological estimation method of textural elements based on the Primitive, Grain, and Point Configuration (PGPC) texture model is proposed. The PGPC texture model has shown promising applications such as noise removal, texture modification, and texture synthesis. However, the estimation is not always successful since the magnification process of the primitive to ensure the assumption that the grains are homothetic does not always fit to each image. We propose in this paper a novel estimation method introducing more flexibility into homotheticity's conventional assumption of the grains, and exploring a suitable structuring element for the homothetic magnification process of the primitive. Experimental results show that the proposed method provides more representative grains than the conventional method.

1 Introduction

Texture analysis is an important and useful area of image processing [1]. Formal definitions of texture vary in the literature; one of the most common describes texture as being generated by one or more basic local patterns that are repeated in a periodic manner over some image region [2]. Mathematical morphology [3,4,5] is a popular method of texture analysis. In our previous study, we proposed a model for texture description based on morphological operations, called the Primitive, Grain, and Point Configuration (PGPC) texture model [6]. The PGPC model represents a texture as an image composed of regular or irregular arrangements of objects, which are much smaller than the image. The arranged objects, called grains, are derived from one or a few typical objects, called primitives. The PGPC texture model is useful in applications such as optimizing nonlinear filters for noise removal and texture modification [7,8,9].

Applying the PGPC texture model to practical texture analysis requires a method of estimating a structuring element as the primitive, its homothetic magnifications as the grains, and morphological skeletons as point configurations.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 190–199, 2011.

In our previous study, experiments with estimating these three parameters under the PGPC model showed promising results [8,10].

In the PGPC texture model, we denote nB as the n-times homothetic magnification of a structuring element B,and is usually defined in the context of mathematical morphology as

$$nB = \begin{cases} B \oplus B \oplus \ldots \oplus B \quad ((n-1) - times \;\; of \;\; \oplus) & (n > 0), \\ \{0\} & (n = 0), \end{cases} \tag{1}$$

where $\{0\}$ denotes a single dot at the origin. The operation $X \oplus Y$ is called Minkovski set addition of the sets X and Y, defined as follows:

$$X \oplus Y = \{x + y | x \in X, y \in Y\}, \tag{2}$$

The PGPC texture model represents a texture image X as

$$X = \bigcup_{n=0}^{N} B_n \oplus \Phi_n, \tag{3}$$

for nonempty Φ_n, where B_n denotes a grain, Φ_n is a point configuration — a set indicating pixel positions for locating the grain nB.

We assume here that $\{0B, 1B, \ldots, nB, \ldots\}$ are homothetic magnifications of a small object B as defined in (1), and that B_n in (3) is equivalent to nB for each n. In this case, B is the primitive, n is the size of the magnification, X_{nB} is regarded as the texture image composed only of the arrangement of nB.

Since the difference between the extents of nB and $(n+1)B$ is often too large in the usual definition in (3), which leads to large gaps between grain sizes. Thus, we redefined the magnification for the PGPC texture model in our previous study as

$$nB = \begin{cases} B \oplus C \oplus \ldots \oplus C \quad ((n-1) - times \;\; of \;\; \oplus) & (n > 0), \\ \{0\} & (n = 0), \end{cases} \tag{4}$$

where C is a 2×2-pixel square SE.

The grain magnification method in (4) provides a convenient way of decreasing the large difference of nB and $(n+1)B$, which make the details of the image finer and less complicated. However, the fixed 2×2 square SE in the conventional PGPC model expands in all four directions of the smaller grain. It does not completely take into consideration individual variations of natural texture images. This basic idea is slightly unnatural and limits the application of the PGPC model.

In this paper, we present a modified version of the conventional PGPC model, which not only focuses on the primitive but also extends the estimation to the magnifications. The proposed method allows a flexible magnification of grains. Experimental results of the proposed method indicate that this modification of is more suitable for real texture analysis.

2 Primitive and Grain Estimation Using Flexible Magnification

We propose an improved primitive and grain estimation method, which replaces the fixed SE C with estimated SEs for the magnifications. In other words, in the iterative estimation process, we estimate the primitive and its magnifications (grains) simultaneously. We redefine nB as follows:

$$nB = \begin{cases} B \oplus M_1 \oplus \ldots \oplus M_n & ((n-1)-times \;\; of \;\; \oplus) \; (n \geq 1), \\ \{0\} & (n = 0), \end{cases} \tag{5}$$

where M_n is a flexible 4-pixel SE for size n magnification.

The proposed method is called "flexible" since the SE for size n magnification is estimated from all possible candidates of 4-pixel shapes instead of a fixed 2×2 square in the conventional method. The conventional 2×2 square shape provides even extensions in the horizontal, vertical and diagonal directions, while the proposed method enhances only the optimal extension trend and, therefore, estimates more representative grains.

The estimation method explores the SEs that minimizes an evaluation function of the fitness of SE by using simulated annealing. The estimation procedure is:

1. Define an initial primitive B, a maximum size of magnification N and a database containing 51 candidates of SE for magnification. The candidates in the database covers all possible shapes of a 4-pixel, 8-connected convex region.
2. Define Q_N as

$$Q_N = \{M_1, \ldots, M_N\}, \tag{6}$$

Q_N is the combination of SEs for magnification where I_N is minimized,

$$I_N = \sum_{n=0}^{N} (1 - \frac{A\{X_{nB}\}}{A\{X\}}), \tag{7}$$

where A is the area of the operand if the operand is a binary image, and the sum of pixel values if the operand is a grayscale image. X_{nB} denotes the opening of image X by structuring element B of size n magnified by $M_1 \ldots M_n$. Q_N is selected from all the possible combinations of candidates of SE.

3. Assign a modification of B to B', where one pixel of B is randomly selected and removed, and one pixel is added at an adjacent position which is also randomly selected.
4. Calculate Q'_N and I'_N similarly as Step 2.
5. Compare I_N and I'_N.

- If $I_N > I'_N$, replace B with B', and go back to Step 3.

- If $I_N \leq I'_N$, preserve B with a probability $P(I_N, I'_N)$ in order to avoid reaching a local minimum, and go back to Step 3. $P(I_N, I'_N)$ is defined as follows:

$$P(I_N, I'_N) = \frac{1}{1 + \exp(\frac{I'_N - I_N}{T(i)})}, \tag{8}$$

where i is the iteration index and $T(i)$ is called "temperature" at the ith iteration, defined as follows:

$$T(i) = 100 \times (0.98)^i. \tag{9}$$

It decreases along the progress of iteration.

6. Iterate Steps 3–5 by a sufficient number of times. B is the optimized primitive and Q_N are the optimized SEs for grain magnifications.

3 Experimental Results

We used three 8-bit, 128 × 128 texture images [11], shown in Fig. 1 in the experiments. The extent of primitive is fixed to 9 pixels in the following experiments. Figure 1 (a) (straw) has strong diagonal directionality; Figure 1 (b) (brick) has a trend of horizontal extension when the grain size increases; Figure 1 (c) (pellets) contains isotropic disk-like grains.

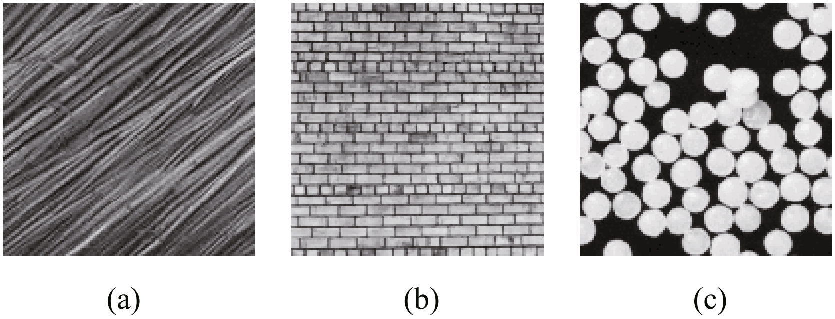


Fig. 1. Target textures used in the experiments: (a) straw, (b) brick, and (c) pellets

A comparison between the proposed and conventional methods [7], shown in Figs. 2–6, confirms that the proposed method provides more representative primitives and grains than the conventional method.

As shown in Fig. 2, the proposed method successfully estimated the directional primitive and its diagonal extension trend. The grains obtained by magnification of flexible SEs closely resemble the original structures of Fig. 1 (a); however, the grains derived by the conventional method became "fatter" after magnification, and failed to preserve the original structure (Fig. 3).

Figure 1 (b) contains two kinds of bricks: nearly square shape and horizontal rectangular. The proposed method estimated a square primitive and two rectangular grains (Fig. 4), while the conventional method only estimated square shapes, which do not resemble all the brick shapes (Fig. 5). Note that in Fig. 4 (h), there are significant dark strips between the horizontal bricks because the grains of size-three do not fit in the smaller, nearly square bricks. However, the estimated square primitive can represent those smaller bricks.

All the grains of Fig. 1 (c) are disk shaped. Under our experimental condition, the magnifications by 2 × 2 square SE had better fitness than other shapes. Thus, the proposed method has exactly the same experimental results as the conventional method (Fig. 6).

The experimental results show that the opening image qualities obtained by the proposed method are generally higher than those provided by the conventional method. From the viewpoint of visual perception, the opening image by the conventional method is quite blurred due to distortion of the grain shape distorted by the fixed 2×2 square SE.

The size distribution functions $F(n)$ of the proposed method and the conventional method are compared in Table 1, which shows the fitness of the primitives and grains to the original texture. A smaller $F(n)$ value indicates higher fitness.

Table 1. Comparison of the size distribution functions $F(n)$ between the proposed method and the conventional method

	proposed method			conventional method		
	primitive (size-one)	grain one (size-two)	grain two (size-three)	primitive (size-one)	grain one (size-two)	grain two (size-three)
straw	**0.0652**	**0.1026**	**0.1527**	0.0910	0.1957	0.2822
brick	0.0435	0.1081	**0.1637**	0.0435	**0.1069**	0.3427
pellets	0.0131	0.0235	0.0361	0.0131	0.0235	0.0361

4 Application to Noise Removal

The proposed method can be applied to the optimization of morphological filters for noise removal. Experiments for impulsive noise removal are shown in the following in order to show experimentally the robustness of the proposed method for the degradation of images in the primitive estimation. We employ the case of impulsive noise since morphological filtering is known to be suitable for such noise.

In our previous research [6,7,8,9,10,12], we adopted only the estimated primitive as the SE for noise removal. However, any noise removal method is a trade-off between preserving the signal and discarding the noise. When the noise ratio is high, a smaller SE can retain more information; however, the efficiency of noise removal is reduced. In this case, use of a larger grain might be more efficient than the primitive.

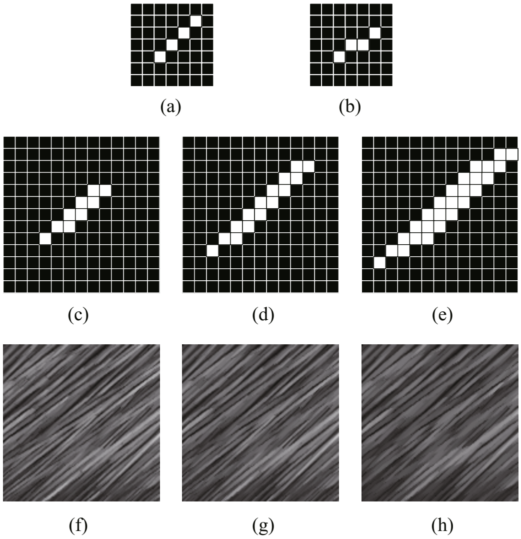


Fig. 2. Experimental results of the proposed method with a straw texture: (a) selected SE for size-two magnification, (b) selected SE for size-three magnification, (c) estimated primitive, (d) size-two magnification (grain one), (e) size-three magnification (grain two), (f) opening of the original texture by (c), (g) opening of the original texture by (d), and (h) opening of the original texture by (e)

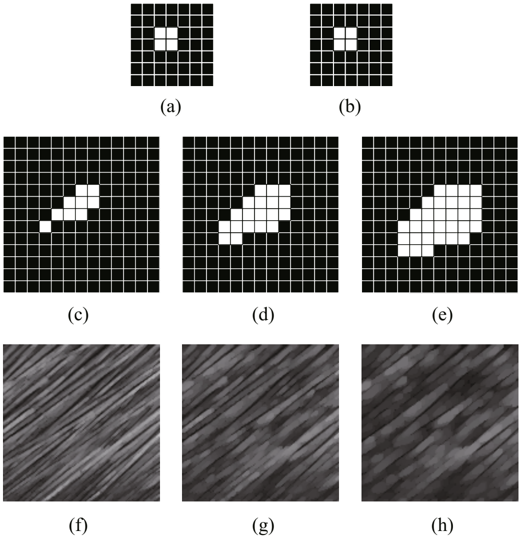


Fig. 3. Experimental results of the conventional method on a straw texture: (a) and (b) square SE for size-two and three magnifications, (c) estimated primitive, (d) size-two magnification (grain one), (e) size-three magnification (grain two), (f) opening of the original texture by (c), (g) opening of the original texture by (d), and (h) opening of the original texture by (e)

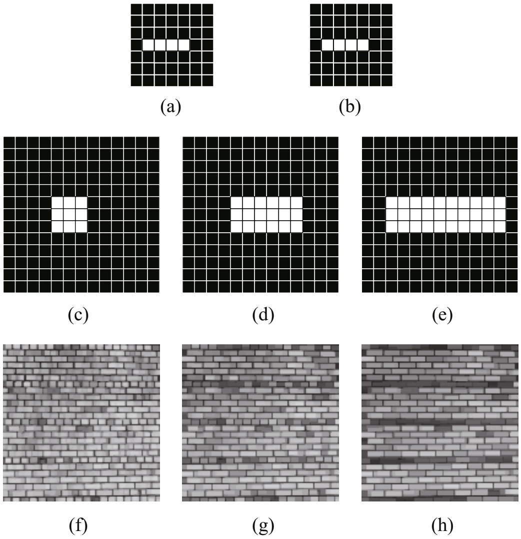


Fig. 4. Experimental results of the proposed method on a brick texture: (a) selected SE for size-two magnification, (b) selected SE for size-three magnification, (c) estimated primitive, (d) size-two magnification (grain one), (e) size-three magnification (grain two), (f) opening of the original texture by (c), (g) opening of the original texture by (d), and (h) opening of the original texture by (e)

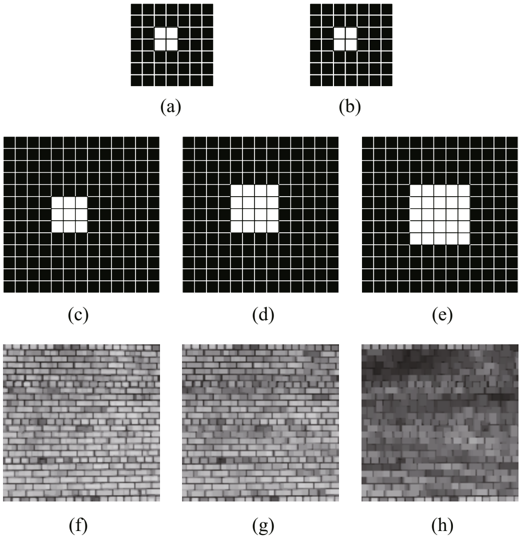


Fig. 5. Experimental results of the conventional method on a brick texture: (a) and (b) square SE for size-two and three magnifications, (c) estimated primitive, (d) size-two magnification (grain one), (e) size-three magnification (grain two), (f) opening of the original texture by (c), (g) opening of the original texture by (d), and (h) opening of the original texture by (e)

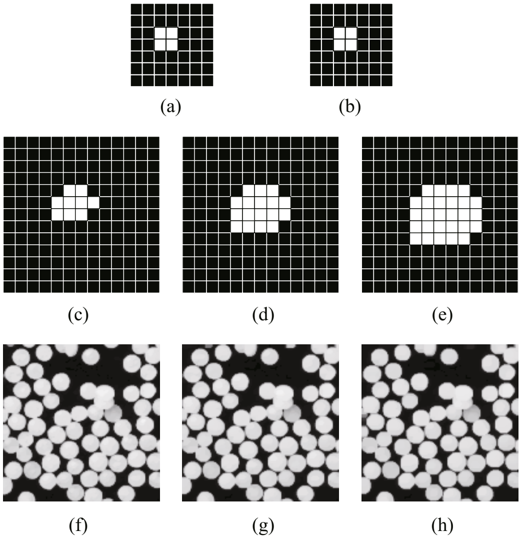


Fig. 6. Experimental results of the proposed method on a pellet texture: (a) selected SE for size-two magnification, (b) selected SE for size-three magnification, (c) estimated primitive, (d) size-two magnification (grain one), (e) size-three magnification (grain two), (f) opening of the original texture by (c), (g) opening of the original texture by (d), and (h) opening of the original texture by (e)

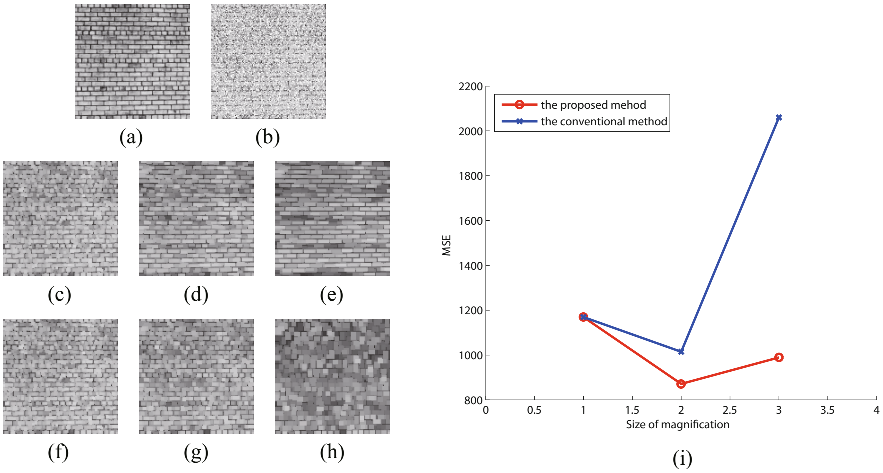


Fig. 7. An example of noise removal on texture "brick": (a) original image, (b) corrupted image, (c)-(e) noise removal by three magnifications of the proposed method, (f)-(h) noise removal by three magnifications of the conventional method, and (i) MSE values

Figure 7 shows an example of noise removal using the estimated primitive and grains on a brick image with a noise ratio of 60%. In this case, the size-two magnification (grain one) is more suitable for noise removal. The proposed method exhibits better noise removal performance than the conventional method — the proposed method preserves more textural details while removing the high rate noise in the target image. We will extend the proposed algorithm by employing our method of multiple SE optimization [12] for multiprimitive textures.

5 Conclusions and Discussion

The parameter estimation methods [6,7,8,9,10,12] only focused on the primitive estimation and applied a 2 × 2 square SE for all magnifications. The improved PGPC model, proposed in this paper, estimates the primitive and the grains simultaneously by selecting the best SEs for each magnification. The experimental results show the effectiveness of the proposed method.

This improved model is easily utilized for applications of noise removal and texture reconstruction. However, the extent of primitive is fixed to nine pixels in the current study and SE is estimated from the original images. We will try the experiments to estimate SEs directly from corrupted images by the method in [7]. The optimization of primitive size will also be studied in the future.

Acknowledgment

This study was partially supported by a Grant-in-Aid for Scientific Research C (20560357) from the Japan Society for Promotion of Science.

References

1. Petrou, M., Garicia-Sevilla, P.: Image Processing: Deal with Texture. John Wiley & Sons Ltd., Chichester (2006)
2. Wechsler, H.: Texture analysis - a survey. Signal Processing 2, 271–282 (1980)
3. Serra, J. (ed.): Image Analysis and Mathematical Morphology. Technical Advances, vol. 2. Academic Press, London (1988)
4. Soille, P.: Morphological Image Analysis, 2nd edn. Springer, Heidelberg (2003)
5. Heijmans, H.J.A.M.: Morphological Image Operators. Academic Press, London (1994)
6. Asano, A., Ohkubo, T., Muneyasu, M., Hinamoto, T.: Primitive and Point Configuration texture model and primitive estimation using mathematical morphology. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 178–185. Springer, Heidelberg (2003)
7. Fujiki, A., Asano, A., Muneyasu, M.: Unsupervised structuring element optimization of morphological opening for texture images. In: Proc. 2006 International Symposium on Nonlinear Theory and its Applications, pp. 711–714 (2006)

8. Kobayashi, Y., Asano, A.: Modification of spatial distribution in primitive and point configuration texture model. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 877–884. Springer, Heidelberg (2003)
9. Li, L., Asano, A., Muraki Asano, C.: Statistical analysis of human visual impressions on morphological image manipulation of gray scale textures. Optical Review 17(2), 90–96 (2010)
10. Asano, A., Kobayashi, Y., Muraki, C., Muneyasu, M.: Optimization of gray scale morphological opening for noise removal in texture images. In: Proc. 47th IEEE International Midwest Symposium on Circuits and Systems, vol. 1, pp. 313–316 (2004)
11. Brodatz, P.: Textures: A Photographic Album for Artists and Designers. Dover, New York (1966)
12. Li, L., Muraki Asano, C., Asano, A.: Dual primitive estimation for morphological texture analysis. In: Proc. 10th International Symposium on Communications and Information Technologies, pp. 149–152 (2010)

Geodesic Attributes Thinnings and Thickenings

Vincent Morard, Etienne Decencière, and Petr Dokladal

CMM-Centre de Morphologie Mathématique
Mathématiques et Systèmes, MINES ParisTech;
35, rue Saint-Honoré, 77305 Fontainebleau CEDEX - France
{Vincent.Morard,Etienne.Decenciere,Petr.Dokladal}@mines-paristech.fr

Abstract. An attribute opening is an idempotent, anti-extensive and increasing operator that removes, in the case of binary images, all the connected components (CC) which do not fulfil a given criterion. When the increasingness property is dropped, more general algebraic thinnings are obtained. We propose in this paper, to use criteria based on the geodesic diameter to build algebraic thinnings for greyscale images. An application to the extraction of cracks is then given to illustrate the performance of the proposed filters. Finally, we will discuss the advantages of these new operators compared to other methods.

Keywords: Geodesic attributes, diameter, elongation, tortuosity, thinnings, thickenings.

1 Introduction

A pre-processing step consists of filtering out the noise and the unwanted features while preserving, as much as possible, the desired information. Mathematical morphology [6], [9] is based on a set approach and classically uses structuring elements (SE) to obtain information on the morphology of the objects. In [10] and in [8], an overview of morphological filtering is presented. We notice that simple openings and closings with square, disk or hexagon SEs, are often good enough for the filtering task. However, if the structuring elements are able to adapt their shapes and sizes to the image's content, the noise reduction and feature enhancement properties are even better (see for example [2] and [3]). The openings and closings by reconstruction can also be considered as a part of adaptive morphology. This leads Vincent [13] to propose area openings, and more generally, Breen and Jones to introduce attributes openings [4].

Here, we start from Lantuéjoul and Maisonneuve's work, [5], to introduce new attributes based on the geodesic diameter. These attributes are particularly useful to measure the length of thin structures. Many papers provide methods to extract thin structures: morphological top hats, supremum of openings by segments, path openings [11] but none of them has the flexibility of the method proposed here.

This work is a part of an industrial project where our goal is to highlight all the defects from metallic surfaces. These cracks are usually long, narrow and

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 200–211, 2011.

not necessarily straight. Standard filters often fail to extract them and these new operators have been developed for this task. More generally, the framework of this study is the detection of cracks.

This paper provides the background to construct algebraic thinnings based on geodesic attributes. Sections 2 and 3 are a review of attribute thinnings and geodesic binary attributes. Section 4 explains how to construct geodesic attribute thinnings, whereas section 5 highlights some practical considerations. Lastly, section 6 illustrates their interest through an application.

2 Background: Attribute Thinnings

2.1 Connected Components and Attributes

Let $I : D \rightarrow V$ be a binary image, with $D \subseteq \mathcal{Z}^2$ typically being a rectangular domain and V the set of values: $V = \{0;1\}$. The object X included in I is $X = \{x \in D | I(x) = 1\}$ and we denote X^c, the complementary set. We associate to I, a local neighbourhood describing the connexion between adjacent pixels. In this study, each pixel will be connected to its eight nearest neighbours. With this 8-connectivity, we define $\{X_i\}$ the set of the connected components of X.

An attribute operator is defined for all connected components X_i in the following way:

$$Att_\lambda(X_i) = \begin{cases} X_i & \text{if } X_i \text{ satisfies } C_\lambda, \\ \emptyset & \text{otherwise.} \end{cases} \tag{1}$$

with C_λ, a criterion parameterised by λ.

2.2 Attribute Thinnings

On the basis of the definition of the attribute operator, a filter ρ^{Att_λ}, called an attribute thinning, can be introduced:

$$\rho^{Att_\lambda}(X) = \bigcup \{Att_\lambda(X_i), i \in I\} \tag{2}$$

Attribute thinnings are anti-extensive and idempotent (see [4] for the proof). Moreover, if these operators are also increasing, they become attribute openings, denoted γ^{Att_λ}.

The dual transform of a thinning is called a thickening and is defined by duality. In what follows, we restrict our study to thinnings as the computation of thickenings is straightforward.

The non increasingness of these filters could cause some problems, especially when we compute granulometries, ultimate thinnings or greyscale thinnings. In the literature, some solutions have been proposed to solve these issues and we will discuss this point when we will extend these operators to grey level images.

2.3 Grey Level Operator

The extension of attribute thinnings to grey level images is not straightforward, since these operators are not always increasing. First, we will talk about the classical method for openings. Then, we will describe the procedure for thinnings.

The openings extend to the grey level domain in the usual way, by thresholding the initial image to obtain N binary images (with N the number of grey levels in the image). Thus, an opening in grey level may be constructed explicitly by stacking the result of each binary opening, computed from each threshold of the original image. With f an image, $f : D \to V$ with $V = \{0, \ldots, N\}$, the grey level attribute opening is given by:

$$(\gamma^{Att_\lambda}(f))(x) = \sup\left\{h \in \{0, \ldots, N\} \left| x \in \gamma^{Att_\lambda}(T_h(f))\right.\right\} \tag{3}$$

where $T_h(f)$ stands for the threshold of f at value h. Throughout this paper, this method will be referred as the "opening binary to grey extension" (OBGE).

For thinnings, at least two methods are available in the literature to construct this extension. They are both presented by Breen and Jones, in [4].

The first one is a method which preserves the information for lower threshold values, once the criterion is fulfilled. Then, it locates the threshold set that satisfy the criterion. Hereafter, this extension is referred to as the "thinning binary to grey extension" (TBGE), and this method has the favour of Breen and Jones.

However, another solution consist in applying exactly the same method as for openings (See OBGE equation 3). Figure 1 (curve b), presents the result of a greyscale thinning with the non increasing criterion: *have a length equal to* λ. In this example, this criterion is fulfilled for a high threshold value whereas it is not true for lower thresholds. The one-dimensional signal is truncated and some edges are emphasised. Therefore, we have filtered out all the unwanted information. Regarding curve c, this is a thinning using the TBGE method and it behaves as a morphological reconstruction by dilation of the curve b.

A real example is presented in section 4.2 where we will discuss the consequences of each method on the result.

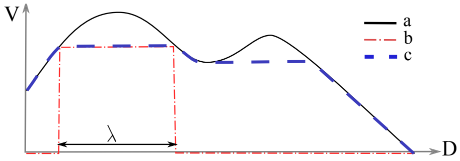


Fig. 1. Example of a grey level thinning with the criterion: *have a length equal to* λ: (a) initial signal, (b) result of the thinning using the OBGE method. (c) result of the thinning using the TBGE method. This curve can be seen as the morphological reconstruction by dilation of the thinned signal (curve b).

3 Geodesic Attributes

From now on, an “object” will be a connected component. The following definitions are valid in continuous or discrete contexts. In practice, as previously said, computations are done in $\mathcal{Z}^2$, with an 8-connectivity. Hereafter, we will define the geodesic attributes on an arbitrary object.

3.1 Geodesic Diameter

Lantuéjoul and Maisonneuve, in [5], asked a question: “What is the length of an object?” The first idea is to measure the length of the segment between its end points (Figure 2(a) and 2(b)); though, it is not a satisfactory definition, as this segment is not always a path included within the object. Moreover, defining the end points is not a trivial question. Another measurement can be considered: the length of the set of points corresponding to a homothetic skeleton of the object (Figure 2(c) and 2(d)). Since this is a part of the object, its length is a more representative measurement. However, skeleton computation methods are difficult to use. Fluctuations can be inserted when small modifications are involved, especially when the objects have rough boundaries.

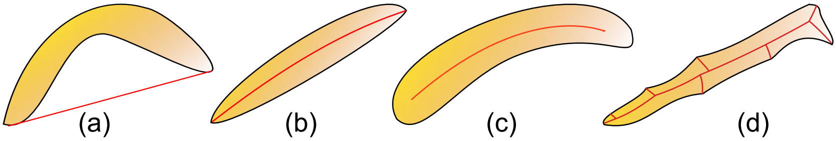


Fig. 2. (a) and (b): the length of the segment between its ends points. (c) and (d), the measurement of its skeleton. These definitions are not always suitable.

Lantuéjoul and Maisonneuve use the notion of geodesic arcs, which are the shortest paths between two points of an object. Let X be an object and x, y two points from X. Figures 3(a) and 3(b) illustrate two paths between these two points and their corresponding geodesic arc, whose length is written $dX(x, y)$. Thus, measuring the length of an object is measuring the length of its longest geodesic arc (Figure 3(c)):

$$L(X) = \sup_{x,y \in X} dX(x, y) \tag{4}$$

$L(X)$ is the geodesic diameter of X and has mainly three advantages: it is a general definition, as it is valid for every object. It is a robust definition, as a small change in the shape of the object will cause, at most, a small change of the measure of the geodesic diameter, if the topology of the object is not changed. Finally, the computation of $L(X)$ leads to other attributes such as the geodesic elongation and the geodesic tortuosity.

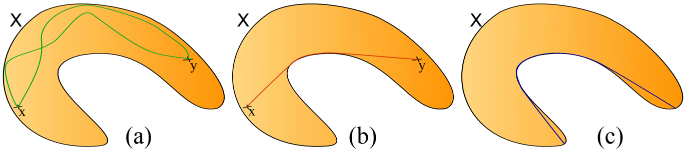


Fig. 3. (a), two paths between x and y; (b), geodesic arc between these two points; (c), geodesic diameter of X

3.2 Geodesic Elongation

The geodesic diameter is the first available attribute and it gives a satisfactory definition of the length of an object. However, we do not have many details on its shape. By combining the length factor with the area of the CC, we do have some information on its tendency to be elongated. The longer and narrower an object is, the higher the elongation factor will be. On the contrary, any disk will have a value of 1. The elongation factor, introduced in [5], is computed as follows:

$$E(X) = \frac{\pi L^2(X)}{4S(X)} \tag{5}$$

where $S(X)$ denotes the area of X. Note that this definition can naturally be generalised to higher dimensions.

3.3 A New Geodesic Attribute: The Geodesic Tortuosity

We propose a new descriptor derived from the geodesic diameter: the geodesic tortuosity. A pair of points $\{x, y\}$ is called a pair of geodesic extremities of X if and only if $dX(x, y) = L(X)$. Note that some objects may have more than one pair of geodesic extremities (ie. a disk). Let $E_x(X) = \{\{x_0, y_0\}, \{x_1, y_1\}, \ldots\}$ be the set of geodesic extremities of X. Then we define $L_{Eucl}(X)$ as the minimal Euclidian distance between geodesic extremities:

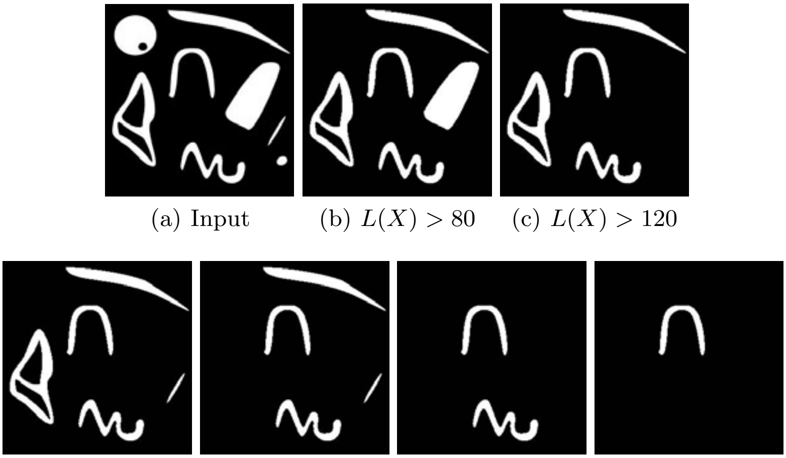
$$L_{Eucl}(X) = \min_{(x,y)\in E_x(X)} \|x, y\| \tag{6}$$

The tortuosity is the ratio between the geodesic diameter and $L_{Eucl}(X)$ (Equation 7). The more twisted the object is, the higher its tortuosity will be. On the contrary, any straight object will be valuated by 1.

$$T(X) = \frac{L(X)}{L_{Eucl}(X)} \tag{7}$$

3.4 Geodesic Attribute Properties and Comments

All these attributes are rotation invariant. Moreover, the geodesic elongation and tortuosity attributes are scale invariant. Other attributes could be derived from the computation of the geodesic diameter; we can name one, which is



(a) Input (b) $L(X) > 80$ (c) $L(X) > 120$

(d) $E(X) > 5$ (e) $E(X) > 10$ (f) $T(X) > 1.5$ (g) $T(X) > 2$

Fig. 4. Filtering result with geodesic attributes criteria C_λ: (a) initial image; (b) and (c): geodesic diameter; (d) and (e): geodesic elongation; (f) and (g): geodesic tortuosity

scale and rotation invariant: the circularity attribute. It is, in fact, the inverse of the geodesic elongation. In comparison, Urbach and Wilkinson, in [12] and [14], used the moment of inertia instead of the geodesic diameter to compute the elongation attribute. However, they are two very different attributes and the geodesic diameter is, in our opinion, a better representation of the length of an object.

4 Geodesic Attributes Thinnings

The main idea of this paper is to combine geodesic attributes with thinnings to obtain a new powerful family of filters.

4.1 Binary Images

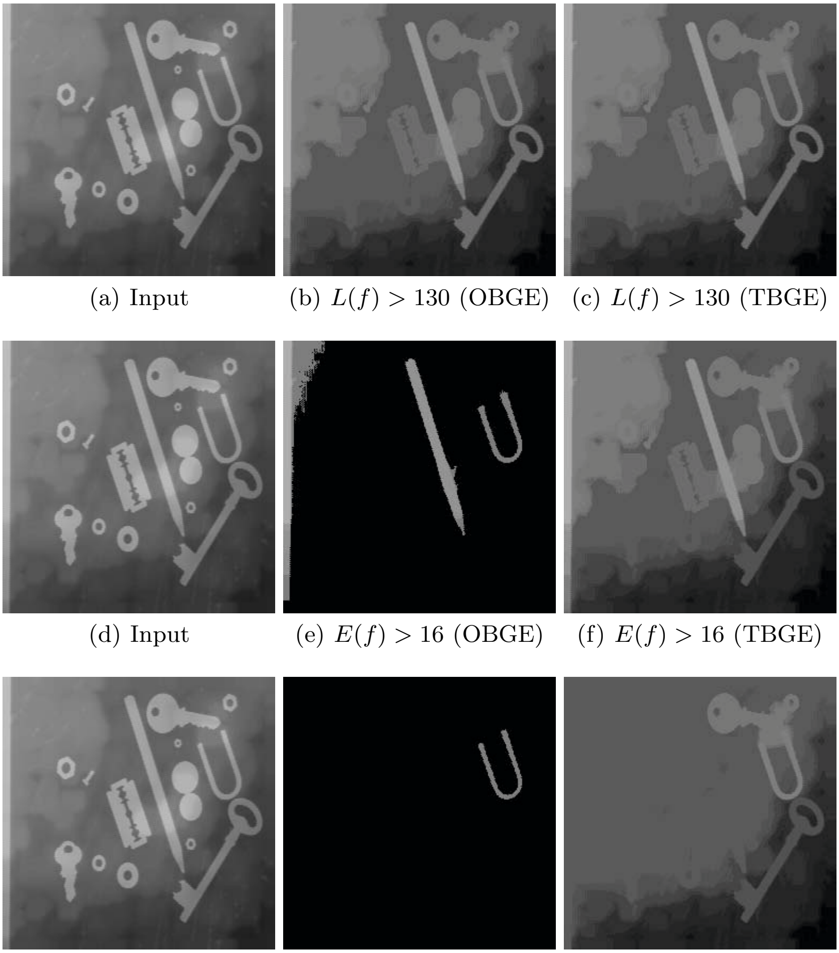
Figure 4(a) is a toy example where we can apply our different operators. It is a set of objects which look like fibres and we want to filter out these objects, with the following criteria:

– Suppress the particles whose geodesic diameter is smaller than 80 pixels in Figure 4(b) and smaller than 120 pixels in Figure 4(c);
– Suppress the particles which are not elongated, i.e. whose geodesic elongation is smaller than 5 in Figure 4(d) and smaller than 10 in Figure 4(e);
– Suppress all particles which are not tortuous, i.e. whose geodesic tortuosity is smaller than 1.5 in Figure 4(f) and smaller than 2 in Figure 4(g).

Hence, we can characterise the shape of these structures with a good accuracy.

4.2 Grey Level Images

Two methods have been presented in section 2.3, to extend this algorithm to grey level images. Figure 5 shows the differences between these approaches. For a segmentation task, using the OBGE method, yields the best results (Figures 5(b), 5(e) and 5(h)) since the tools are correctly isolated from the background and a simple threshold often leads to the wanted segmentation. However, to isolate all the filtered objects of the image, a top hat has to be computed on the thinned image using the TBGE method.



(a) Input (b) $L(f) > 130$ (OBGE) (c) $L(f) > 130$ (TBGE)

(d) Input (e) $E(f) > 16$ (OBGE) (f) $E(f) > 16$ (TBGE)

(g) Input (h) $T(f) > 6$ (OBGE) (i) $T(f) > 6$ (TBGE)

Fig. 5. Filtering by geodesic attributes: diameter, elongation and tortuosity. The first column is the initial image. For the second (resp. third) column, the extension for greyscale image is the OBGE method (resp. TBGE method)), discussed in 2.3.

We notice for the geodesic diameter, these two methods give exactly the same result. Thus, this thinning based on the geodesic diameter behaves as an opening for this image. This is due to the fact that most of the objects in figure 5(a) have a convex shape.

The choice of the method will depend on our applications. In the following, the OBGE method is used, as we want to isolate the cracks from the background.

5 Practical Considerations and Optimisation

5.1 Computation of the Geodesic Diameter

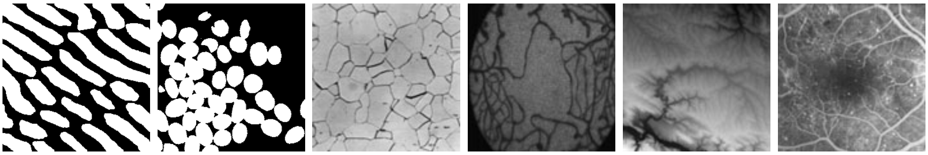
The geodesic diameter has to be computed for every CC of every threshold of the image. Hence, the complexity of this algorithm depends mainly on the number and the area of these CCs. Maisonneuve and Lantuéjoul in [5] designed an efficient parallel implementation for binary images to compute the geodesic diameter in a hexagonal grid. Let Y be a set of emitting sources and X be an object simply connected with Y a subset of X. The computation of $L(X)$ requires a propagation step from the emitting sources into X. If Y is correctly chosen (e.g. the boundaries of the objects), the geodesic diameter would be deduced from the last wave iteration. However, this algorithm does not support holes; the CC has to be simply connected, otherwise the propagation wave would never end, turning infinitely around the holes. This is a real limitation to the use of this algorithm.

Classical attribute filters are often based on a tree representation, as presented by Salembier et al. in [7]. However, we could not find a fast way to update the geodesic diameter value, when a new pixel is added to a CC. Hence, the connected component tree representation is not as efficient as for simple attributes (area, width, height).

Then, we prefer using a direct implementation where the greyscale image is converted into binary images. Each connected component is isolated using a stack of pixels as container. Each pixel belonging to the boundary of the object is a starting point to a region growing process in order to build a distance map. The highest value of all the distance maps is the geodesic diameter.

5.2 Optimisation

A possible acceleration is available for the geodesic diameter and the geodesic elongation thinnings. During, the region growing process, when the front wave becomes larger than the attribute, it is useless to compute the real value of this attribute. The criterion is passed, we keep the current connected component and we can stop the propagation step. The time saved is huge (see table 1 and figure 6) but it does depend on λ.



(a) Eutectic (b) Coffee (c) Grains3 (d) Macula (e) Relief (f) Retina2

Fig. 6. Images used to build table 1

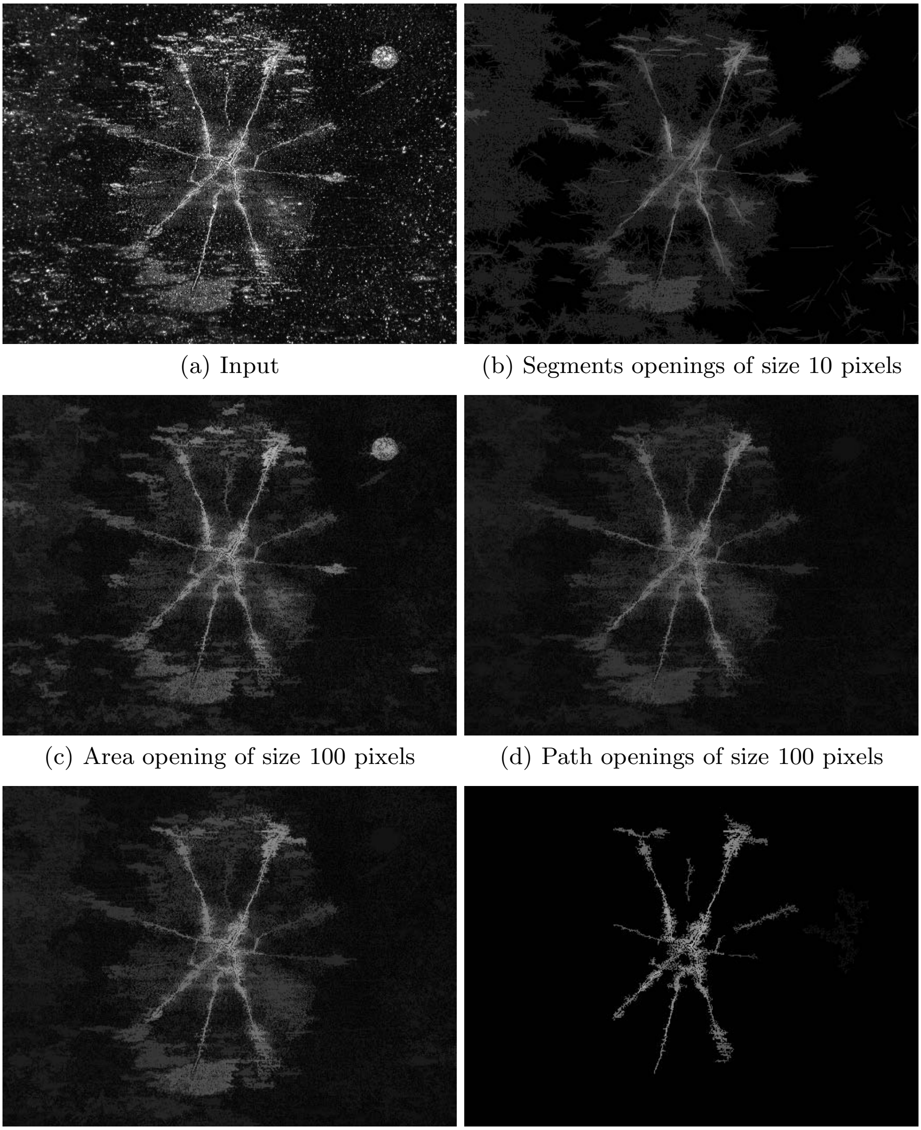
Table 1. Running times for different images for a geodesic diameter thinning or thickening with the OBGE method and for $\lambda = 20$. Timings are in seconds. Laptop computer: Intel Core2 Duo T7700 @ 2.40GHz

Images	Direct method	Accelerated method (see 5.2)
Coffee (256 x 256 x 1-bit)	36	0.079
Eutectic (256 x 256 x 1-bit)	37	0.015
Grains3 (256 x 256 x 8-bits)	3650	5.4
Macula (256 x 256 x 8-bits)	3850	2.7
Relief (256 x 256 x 8-bits)	1024	0.99
Retina2 (256 x 256 x 8-bits)	1820	1.56

6 Results

The new operators have been applied to real images in the framework of our project. Here, the geodesic attribute thinnings are used to detect long and narrow structures. The proposed image, is a crack and we want to extract it (Figure 7(a)). We use and compare five different methods in order to do it:

- The supremum of openings by segments of size 10 pixels oriented every 2 degrees. Figure 7(b) presents the result and we see that, only the linear part of the crack is preserved. This method is used to extract linear features, and when a crack is not straight, this method is not efficient.
- An area opening of size 100 pixels (Figure 7(c)). Here we observe that the noise is correctly filtered out. However, the circle structure is preserved, as well as the compact noise which is larger than 100 pixels.
- A maximum path opening of size 100 pixels (Figure 7(d)). For each pixel, the graph has 3 predecessors and 3 successors according to [1]. The result is much better than the previous method. However, not all the branches of the crack are extracted. When the path is too tortuous, this algorithm is not able to follow the entire crack and fails to estimate its length.
- A geodesic diameter thinning of size 100 pixels yields a better result since all the branches are correctly extracted (Figure 7(e)). This algorithm is a connected operator. The tortuosity of the CCs has no influence over its length.
- A geodesic elongation thinning of value 20 (Figure 7(f)). This method filters out all the noise and offers a very efficient detection.


(a) Input

(b) Segments openings of size 10 pixels

(c) Area opening of size 100 pixels

(d) Path openings of size 100 pixels

(e) Geodesic diameter thinning of size 100 pixels

(f) Geodesic elongation thinning of size 20

Fig. 7. Crack detection: to detect these thin structures, we use 5 different methods. The geodesic attribute thinning yields the best detection.

7 Conclusion and Future Work

We have presented new attributes thinnings based on geodesic criteria. The geodesic elongation, the geodesic diameter and the geodesic tortuosity are non increasing criteria which offer good filtering capabilities. Thus, the extraction of long and elongated structures is easy and is made in an efficient way. It offers more flexibility compared to other methods. Moreover, we can have a representation of the elongation and the tortuosity, which is not possible with path openings. An acceleration is proposed for thinnings based on the geodesic diameter and the geodesic elongation. Therefore, these operators are fast enough for many applications.

Speed up the computation of the geodesic diameter seems to be difficult. However, we are working on the elaboration of a new strategy to approximate the geodesic diameter with a high accuracy. In practice, we get very similar results but the final algorithm is several time faster. In average, the running times are divided by a factor of 20, compared to the accelerated method presented in 5.2. Hence, the extension to 3D images will be straightforward. Future work will also include granulometries and ultimate thinnings with geodesic attributes.

Acknowledgments

This work has been financed by the French "département de Seine et Marne". The authors are grateful to Ms Raviart, working in the "Centre des matériaux, MINES ParisTech", for her help with the optical microscope.

References

1. Appleton, B., Talbot, H.: Efficient path openings and closings. Mathematical Morphology: 40 Years On, pp. 33–42 (2005)
2. Beucher, S., Blosseville, J.M., Lenoir, F.: Traffic spatial measurements using video image processing. In: Intelligent Robots and Computer Vision. Proc. SPIE, vol. 848, pp. 648–655 (1987)
3. Braga-Neto, U.: Alternating sequential filters by adaptive-neighborhood structuring functions. Mathematical Morphology and its Applications to Image and Signal Processing, 139–146 (1996)
4. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. Computer Vision and Image Understanding 64(3), 377–389 (1996)
5. Lantuéjoul, C., Maisonneuve, F.: Geodesic methods in quantitative image analysis. Pattern Recognition 17(2), 177–187 (1984)
6. Matheron, G.: Random sets and integral geometry, vol. 9. Wiley, New York (1975)
7. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. IEEE Transactions on Image Processing 7, 555–570 (1998)
8. Salembier, P., Wilkinson, M.H.F.: Connected operators. IEEE Signal Processing Magazine 26(6), 136–157 (2009)
9. Serra, J.: Image analysis and mathematical morphology. Academic, London (1982)
10. Serra, J., Vincent, L.: An overview of morphological filtering. Circuits Syst. Signal Process. 11(1), 47–108 (1992), http://portal.acm.org/citation.cfm?id=150488
11. Talbot, H., Appleton, B.: Efficient complete and incomplete path openings and closings. Image and Vision Computing 25(4), 416–425 (2007)

12. Urbach, E., Wilkinson, M.: Shape-only granulometries and gray-scale shape filters. In: Mathematical Morphology: Proceedings of the VIth International Symposium, ISMM 2002, p. 305. Csiro (2002)
13. Vincent, L.: Grayscale area openings and closings, their efficient implementation and applications. In: First Workshop on Mathematical Morphology and its Applications to Signal Processing, pp. 22–27 (1993)
14. Wilkinson, M.H.F., Westenberg, M.A.: Shape preserving filament enhancement filtering. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, pp. 770–777. Springer, Heidelberg (2001)

Morphological Bilateral Filtering and Spatially-Variant Adaptive Structuring Functions

Jesús Angulo

CMM-Centre de Morphologie Mathématique, Mathématiques et Systèmes, MINES
ParisTech; 35, rue Saint Honoré, 77305 Fontainebleau Cedex, France
jesus.angulo@mines-paristech.fr

Abstract. Development of spatially-variant filtering is well established in the theory and practice of Gaussian filtering. The aim of the paper is to study how to generalize these linear approaches in order to introduce adaptive nonlinear filters which asymptotically correspond to spatially-variant morphological dilation and erosion. In particular, starting from the bilateral filtering framework and using the notion counter-harmonic mean, our goal is to propose a new low complexity approach to define spatially-variant bilateral structuring functions. Then, the adaptive structuring elements are obtained by thresholding the bilateral structuring functions. The methodological results of the paper are illustrated with various comparative examples.

1 Introduction

Let E be the Euclidean $\mathbb{R}^d$ or discrete space $\mathbb{Z}^d$ (support space) and let $\mathcal{T}$ be a set of grey-levels (space of values). It is assumed that $\mathcal{T} = \overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. A grey-level image is represented by a function $f : E \to \mathcal{T}$, $f \in \mathcal{F}(E, \mathcal{T})$, i.e., f maps each pixel $\mathbf{x} \in E$ into a grey-level value $t \in \mathcal{T}$: $t = f(\mathbf{x})$. Given a grey-level image, the two basic morphological mappings $\mathcal{F}(E, \mathcal{T}) \to \mathcal{F}(E, \mathcal{T})$ are the dilation and the erosion given respectively by $\delta_b(f)(\mathbf{x}) = \sup_{\mathbf{y} \in E} (f(\mathbf{x} - \mathbf{y}) + b(\mathbf{y}))$, and $\varepsilon_b(f)(\mathbf{x}) = \inf_{\mathbf{y} \in E} (f(\mathbf{x} + \mathbf{y}) - b(\mathbf{y}))$; where $b \in \mathcal{F}(E, \mathcal{T})$ is the structuring function which determines the effect of the operator. The other morphological operators, such as the opening and the closing, are obtained as products of dilation/erosion. The most commonly studied framework, which additionally presents better properties of invariance, is based on flat structuring functions, called structuring elements. More precisely, let B be a Boolean set defined at the origin, i.e., $B \subseteq E$ or $B \in \mathcal{P}(E)$, which defines the "shape" of the structuring element, the associated structuring function is given by $b(\mathbf{x}) = 0$ if $\mathbf{x} \in B$ and $b(\mathbf{x}) = -\infty$ if $\mathbf{x} \in B^c$ (where B^c is the complement set of B). Hence, the flat dilation and flat erosion can be computed respectively by the moving local maxima and minima filters.

Standard formulation of morphological operators is translation invariant in the space and in the intensity [11], i.e., the same structuring function $b(\mathbf{x})$ (or structuring element B) is considered for each point $\mathbf{x}$ of the image. A current challenging topic in mathematical morphology is the construction of adaptive

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 212–223, 2011.

operators; i.e., structuring functions become dependent on the position or on the input image itself. For a recent overview on the state-the-art on adaptive morphology, the interested reader is invited to papers [15,18].

Aim of the paper. Development of locally adaptive filtering is well established in the theory and practice of Gaussian filtering. The aim of this paper is to study how to generalize these linear approaches in order to introduce adaptive nonlinear filters which asymptotically correspond to spatially-variant morphological dilation and erosion. In particular, starting from the bilateral filtering and using the notion of counter-harmonic mean, our goal is to propose a new low complexity approach to define spatially-variant structuring functions (Section 2). After thresholding, adaptive structuring elements are obtained which can be used to compute spatially-variant dilation, erosion, opening and closing (Section 3). The associated algorithms take advantage of the low computational complexity of bilateral filtering.

Related work. Bilateral filtering [22] is a locally adaptive Gaussian convolution technique to smooth images while preserving edges, where the Gaussian coefficients at a point are weighted by the intensity distance between its neighbours. A recent systematic study on the theory and applications of bilateral filtering can be found in [17]. In particular, bilateral filtering is a discrete filter equivalent asymptotically to Perona and Malik PDE equation [16,7]. The construction of locally adaptive edge-preserving structuring elements has been previously considered in the literature of mathematical morphology. The notion of generalized geodesy [20] was proposed to introduce locally adaptive geodesic neighbourhoods. Morphological amoebas [12] were proposed as a flexible discrete approach to compute locally adaptive structuring elements. Morphological amoebas have been basically considered for median filtering. In a recent study [28], it has been established an interesting correspondence between the iterated amoeba median filtering and the PDE of self-snakes. More recently, it has been introduced in [10] a framework to compute adaptive kernels using geodesic distances which generalizes the metric of morphological amoebas and adaptive geodesy. But it is exclusively used for averaging edge-preserving smoothing. We should remark that the adaptive neighbourhoods associated to distance propagation in amoebas, or in generalized geodesy, involve a relatively high computational complexity. The starting point of the adopted approach is the notion of counter-harmonic mean [4]. In fact, the idea of using the counter-harmonic mean to construct robust morphological-like operators, without the notions of supremum and imfimum, was proposed in [23]. We have recently used in [1] the counter-harmonic framework to generalize image diffusion in order to introduce iterative nonlinear filters which effects mimic morphological dilation and erosion.

2 Counter-Harmonic Bilateral Filter

Canonic multiscale image analysis involves obtaining the various scales by linear convolutions (i.e., low-pass filtering) of the original image. Hence, each output image pixel value is a weighted sum of its neighbours in the input images, where the weights decrease with the spatial distance to the centre position. These weights are usually given by a Gaussian kernel, which is also related to linear diffusion. As a result, all neighbourhoods are uniformly blurred, including edges and discontinuities.

Bilateral filter: spatially-variant convolution kernels. The rationale behind the bilateral filtering is to apply a weighted average of nearby pixels which depends on the distance but also on the difference in value with respect to the centre. The bilateral filter of image $f(\mathbf{x})$ is defined by the following normalized convolution:

$$\Upsilon(f)(\mathbf{x};\widetilde{k}_{\eta_s,\eta_i}) = \frac{\int_E f(\mathbf{y}) k_{\eta_s}(\|\mathbf{x}-\mathbf{y}\|) k_{\eta_i}(|f_\rho(\mathbf{x}) - f_\rho(\mathbf{y})|) d\mathbf{y}}{\int_E k_{\eta_s}(\|\mathbf{x}-\mathbf{y}\|) k_{\eta_i}(|f_\rho(\mathbf{x}) - f_\rho(\mathbf{y})|) d\mathbf{y}} \qquad (1)$$

where the pair of width parameters defines the vector scale (η_s, η_i), with η_s is the spatial (or size) scale and η_i the intensity (or tonal or range) scale; and where $k_\eta(\xi)$ denotes the function kernel which imposes the smoothing weights of both spatial and contrast effects. By homogeneity, it will be considered that the spatial and intensity kernels are the same. The intensity distances are computed from $f_\rho(\mathbf{x})$, a median-filtered version of the initial image $f(\mathbf{x})$ using a window of size $\rho \times \rho$ pixels. Thus, in the bilateral filter, each neighbour is weighted by a spatial component that penalizes distant pixels and an intensity component that penalizes pixels with a different intensity. The combination by product of both components ensures in $\Upsilon(f)$ that only nearby similar pixels contribute to the final results. The fact of computing the intensity penalization from $f_\rho(\mathbf{x})$, a regularized version of the image, leads to a robust against noise estimator. The latter principle is well known in nonlinear diffusion [7] and in the computation of morphological amoebas (i.e., the "pilot" image [12]). For all the examples given in this paper, we have fixed $\rho = 3$ pixels. This size of median filter (3×3) is appropriate for most of images, including the noisy ones. In cases of severe noise or with very structured images, this parameter could be adaptively selected by choosing a large size, but this issue is beyond the scope of this paper. As in spatially-invariant filtering, increasing the spatial parameter η_s smoothes large features. As the intensity parameter η_i increases, the bilateral filter gradually approximates spatially-invariant filtering, but if the value is too small, no smoothing occurs. In the standard formulation of bilateral filtering, the kernel is a Gaussian function (which provides least squares zero-order estimate of image structure. However, for the nonlinearization case studied in this paper, we propose to consider in detail three alternative kernels:
i) Gaussian kernel: $k_\sigma(\xi) = e^{\frac{-\xi^2}{2\sigma^2}}$;
ii) Laplace kernel $k_\beta(\xi) = e^{\frac{-|\xi|}{\beta}}$;
iii) Cauchy-Lorentz kernel $k_\alpha(\xi) = \frac{1}{1+\frac{\xi^2}{\alpha^2}}$.
Cauchy-Lorentz kernel is an intermediate function between the Gaussian and Laplace kernels, and consequently it is sharper than the Gaussian kernel. Note

also that as the convolution is normalized in the bilateral filtering, the corresponding kernels do not require any additional normalization.

Counter-Harmonic generalization. According to the counter-harmonic paradigm that we have introduced in [1], the following generalized bilateral filter is proposed.

Definition 1. *The counter-harmonic bilateral scale-space of order P using bilateral kernel $\widetilde{k}_{\eta_s,\eta_i}$ is defined as*

$$\Psi(f)(\mathbf{x};\widetilde{k}_{\eta_s,\eta_i};P) = \frac{\int_E f(\mathbf{y})^{P+1} k_{\eta_s}(\|\mathbf{x}-\mathbf{y}\|) k_{\eta_i}(|f_\rho(\mathbf{x}) - f_\rho(\mathbf{y})|) d\mathbf{y}}{\int_E f(\mathbf{y})^{P} k_{\eta_s}(\|\mathbf{x}-\mathbf{y}\|) k_{\eta_i}(|f_\rho(\mathbf{x}) - f_\rho(\mathbf{y})|) d\mathbf{y}}. \quad (2)$$

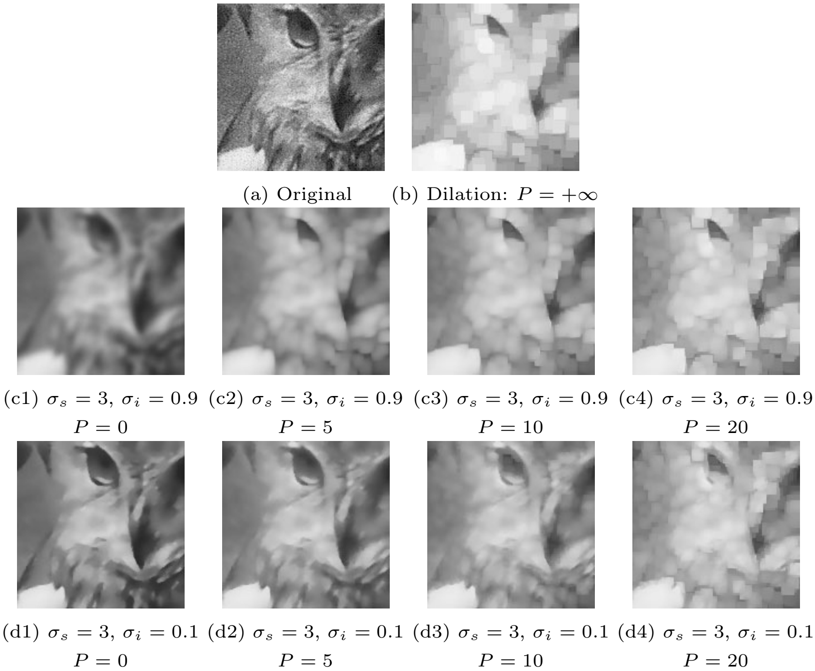
By choosing $P > 0$ (resp. $P < 0$), $\Psi(f)(\mathbf{x};\widetilde{k}_{\eta_s,\eta_i};P)$ leads to a scale-space of pseudo-dilations (resp. pseudo-erosions), which filtering effects for a given pair of spatial/distance scales (η_s,η_i) depend on the "nonlinearity order" of P, which skews the bilateral weighted values towards the supremum or infimum value.

Fig. 1 depicts a comparative example of filtering the image "Owl" using the counter-harmonic bilateral scale-space $\Psi(f)(\mathbf{x};\widetilde{k}_{\eta_s,\eta_i};P)$, using Gaussian kernel, for a fixed spatial scale $\sigma_s = 3$ (windowed in a spatial support of 11×11 pixels). The behaviour of bilateral filtering with respect to the value of σ_i is here well illustrated: with high values, e.g., $\sigma_i = 0.9$, the result is similar to the spatially-invariant Gaussian filter; $\sigma_i = 0.1$ is here a good trade-off to achieve the adaptive effect of bilateral kernels. We observe also the effect of adaptive pseudo-dilations when $P > 0$, for instance with $P = 5$: the spatially-variant dilation results in a moving window max. on regular regions, but without distorting the main edges. We notice that when $P \uparrow\uparrow$, the bilateral pseudo-dilation converges to the spatially-invariant flat dilation ($P = +\infty$), losing the properties of local adaptability. The terminology of "pseudo-dilation", or more generally "pseudo-morphological operators" is probably inappropriate: so many operators have been published as pseudo-morphological ones, without any relation with the present approach. However, due to the fact that the nonlinear filter $\Psi(f)(\mathbf{x};\widetilde{k}_{\eta_s,\eta_i};P)$ for $P > 0$ (resp. $P < 0$) is not extensive (resp. anti-extensive) and does not commute with the supremum (resp. with the infimum), it cannot be considered *stricto sensu* as a dilation (resp. erosion). But let us contemplate in detail the asymptotic behaviour of the bilateral pseudo-dilations/erosion with respect to P.

Limit statements and spatially-variant structuring functions. We can study in particular the limit cases for $P \gg 0$ and $P \ll 0$.

Proposition 1. *For a given pair of scale parameters (η_s,η_i), the limits of (2) with respect to P exist and are given by*

$$\begin{aligned} &\lim_{P\to+\infty} \Psi(f)(\mathbf{x};\widetilde{k}_{\eta_s,\eta_i};P) = \\ &\quad \sup_{\mathbf{y}\in E} \left(f(\mathbf{y}) + \tfrac{1}{P}\left[\log\left(k_{\eta_s}(\|\mathbf{x}-\mathbf{y}\|)\right) + \log\left(k_{\eta_i}(|f_\rho(\mathbf{x}) - f_\rho(\mathbf{y})|)\right)\right]\right) \end{aligned} \quad (3)$$



(a) Original (b) Dilation: $P = +\infty$

(c1) $\sigma_s = 3$, $\sigma_i = 0.9$ $P = 0$ (c2) $\sigma_s = 3$, $\sigma_i = 0.9$ $P = 5$ (c3) $\sigma_s = 3$, $\sigma_i = 0.9$ $P = 10$ (c4) $\sigma_s = 3$, $\sigma_i = 0.9$ $P = 20$

(d1) $\sigma_s = 3$, $\sigma_i = 0.1$ $P = 0$ (d2) $\sigma_s = 3$, $\sigma_i = 0.1$ $P = 5$ (d3) $\sigma_s = 3$, $\sigma_i = 0.1$ $P = 10$ (d4) $\sigma_s = 3$, $\sigma_i = 0.1$ $P = 20$

Fig. 1. Comparison of pseudo-dilations of image $f(\mathbf{x})$ "Owl" using the counter-harmonic bilateral scale-space $\Psi(f)(\mathbf{x}; \widetilde{k}_{\eta_s,\eta_i}; P)$, using Gaussian kernel, for a fixed spatial scale $\sigma_s = 3$ (windowed in a spatial support of 11 × 11 pixels), two intensity scales $\sigma_i = 0.9$, 0.1 and three values of $P > 0$; besides the standard bilateral filtering ($P = 0$) and the flat dilation ($P = +\infty$) of size 11 × 11.

$$\begin{aligned}&\lim\nolimits_{P\to-\infty} \Psi(f)(\mathbf{x}; \widetilde{k}_{\eta_s,\eta_i}; P) = \\ &\qquad \inf\nolimits_{\mathbf{y}\in E} \left(f(\mathbf{y}) - \tfrac{1}{P} \left[\log\left(k_{\eta_s}(\|\mathbf{x}-\mathbf{y}\|)\right) + \log\left(k_{\eta_i}(|f_\rho(\mathbf{x}) - f_\rho(\mathbf{y})|)\right) \right] \right)\end{aligned} \tag{4}$$

Proof is not included by the limited paper length, but it is similar to that given in [1]. Let us consider the asymptotic spatially-variant structuring function:

$$\widetilde{b}(\mathbf{x}; \eta_s, \eta_i; P) = \frac{1}{P} \left[\log\left(k_{\eta_s}(\|\mathbf{x}-\mathbf{y}\|)\right) + \log\left(k_{\eta_i}(|f_\rho(\mathbf{x}) - f_\rho(\mathbf{y})|)\right) \right] = \frac{1}{P} b_{\eta_s,\eta_i}(\mathbf{x}), \tag{5}$$

where $b_{\eta_s,\eta_i}(\mathbf{x}) = b_{\eta_s}(\|\mathbf{x}-\mathbf{y}\|) + b_{\eta_i}(|f_\rho(\mathbf{x}) - f_\rho(\mathbf{y})|)$. Hence, we observe that the supremal convolution of Rel.(3) is exactly a spatially-variant dilation and the infimal convolution of Rel.(4) is a spatially-variant erosion, where the structuring function has been transposed.

This logarithmic connection, which appears here, between the linear and the morphological system theory has been previously considered in the literature [5]. We notice also that, in these limiting cases, the counter-harmonic bilateral framework involves a "normalization" by P of the original spatial and intensity kernel scale parameters during unlinearization. This result is perfectly coherent with those obtained from totally different paradigms [9,27]. We notice that for $P = +\infty \Rightarrow \widehat{\sigma_s} = +\infty$, i.e., the structuring function becomes flat, and hence we obtain the flat dilation in a windowed spatial support. The classical case of the parabolic structuring function as the morphological equivalent of the Gaussian kernel is particularly known in the state-of-the art of mathematical morphology: on the one hand, the parabolic (or more generally, quadratic) structuring functions contain the unique rotationally symmetric structuring functions that can be dimensionally decomposed with respect to the dilation [2]; on the other hand, parabolic structuring functions are the eigenfunctions, which preserve their shape, of the slope transform [8,13], which is the equivalent of Fourier transform in the framework of mathematical morphology. However, to our knowledge, this is the first time that the bilateral parabolic structuring, i.e.,

$$\widetilde{b}_{\sigma_s,\sigma_i}(\mathbf{x}) = -\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma_s^2} - \frac{|f_\rho(\mathbf{x}) - f_\rho(\mathbf{y})|^2}{2\sigma_i^2}, \tag{6}$$

appears in the literature. As we have already stated, the "normalization" introduced by P in the counter-harmonic bilateral filtering involves a flattening of the structuring function $\widetilde{b}(\mathbf{x};\eta_s,\eta_i;P)$ when $|P|$ increases and consequently, the values of pseudo-dilation and pseudo-erosion for $P \gg 0$ and $P \ll 0$ tends respectively asymptotically to a flat dilation and a flat erosion, which lose the properties of adaptability.

At this point, we have the ingredients to carry on the study on bilateral morphology by considering directly the dilation and erosion using the spatially-variant morphological structuring functions, without the resource to the counter-harmonic framework. Nevertheless, this alternative would immediately arrive to a fundamental obstacle: how to calculate the reciprocal dilation of a given erosion, in order to obtain the morphological opening/closing (as product of erosion and dilation), which is a key point in spatially-variant morphology [19]. Hence, we prefer to continue our work following an alternative approach.

3 Bilateral Flat Morphological Operators

The purpose of this section is to introduce the construction of flat spatially-variant morphology using thresholded adaptive structuring functions and a theoretically appropriate formulation of morphological operators.

Thresholding adaptive structuring function. By thresholding $\widetilde{b}_{\eta_s,\eta_i}$ at a fixed value τ, a neighbourhood shape is then obtained for each point $\mathbf{x}$. The binary set $B(\mathbf{x})$, defining an adaptive structuring element, corresponds to a truncated disk. Instead of using η_s as the input parameter to define the spatial scale of the structuring element, we can fix a constant threshold, for instance $\tau = -1$, and then calculate η_s for a desired diameter of the disk d. To interpret the penalization term, we can introduce a new parameter called critical contrast percentage $\varDelta$ (%). This parameter is the value of $|f_\rho(\mathbf{x}) - f_\rho(\mathbf{z})|$ producing

$b_{\eta_i}(\mathbf{x}) = -1$ for a certain η_i, which can be therefore calculated. In summary, we have the following definition.

Definition 2. *Given an image $f(\mathbf{x})$, the adaptive isotropic structuring element at pixel $\mathbf{x}$ of diameter d and critical contrast percentage Δ is defined by*

$$B_{d,\Delta}(\mathbf{x}) = \{\mathbf{z} \in E \mid b_{\eta_s}(\|\mathbf{x}-\mathbf{z}\|) + b_{\eta_i}(|f_\rho(\mathbf{x}) - f_\rho(\mathbf{z})|) \geq -1\}, \qquad (7)$$

where the corresponding spatial and intensity scale parameters for the morphological Gaussian, Laplace and Cauchy-Lorentz kernels are respectively given by

$$\sigma_s = \frac{d}{2\sqrt{2}};\ \beta_s = \frac{d}{2};\ \alpha_s = \frac{d}{2\sqrt{1.718}};\ \textit{and}\ \sigma_i = \frac{1}{\sqrt{2}}\frac{\Delta}{100};\ \beta_i = \frac{\Delta}{100};\ \alpha_i = \frac{1}{\sqrt{1.718}}\frac{\Delta}{100}.$$

Spatially-variant mathematical morphology. In mathematical morphology, when the variation of a structuring function follows a law based on the data, such as in our case where: $\mathbf{x} \to f(\mathbf{x}) \to B_{d,\Delta}(\mathbf{x})$, the complement of the dilation and of the adjoint opening cannot be theoretically calculated [19,25]. Then the four basic operations must be expressed by using the datum of the structuring function $B(\mathbf{x})$ exclusively, without resorting to complement, or equivalently, to reciprocal dilation. For the binary case, dilation and erosion are operations that map a lattice $\mathcal{P}(E)$ into itself preserving both union and intersection. Since a set X is the union of its singletons $\mathbf{x}$, i.e. $X = \cup\{\{\mathbf{x}\}|\{\mathbf{x}\} \subseteq X\}$, and since dilation commutes under union, this operation can be expressed as

$$\delta_{B(\mathbf{x})}(X) = \cup\{B(\mathbf{x})|\mathbf{x} \in X\}, \qquad (8)$$

where $B(\mathbf{x})$ is the structuring element at point $\mathbf{x}$. The expression of the erosion can be drawn from adjunction as

$$\varepsilon_{B(\mathbf{x})}(X) = \{\mathbf{z}|B(\mathbf{z}) \subseteq X\}. \qquad (9)$$

The analytical representation of dilation δ by means of the structuring function extends to the associated opening and closing, defined as

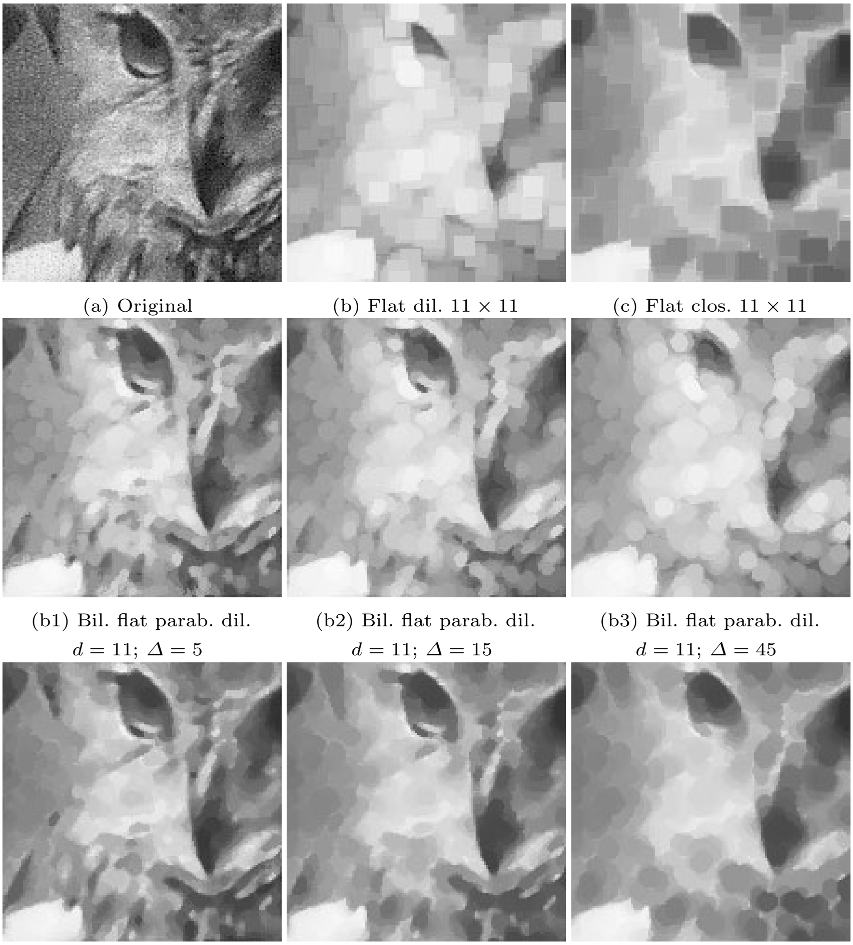
$$\gamma_{B(\mathbf{x})}(X) = \cup\{B(\mathbf{x})|B(\mathbf{x}) \subseteq X\} \qquad (10)$$

and

$$\varphi_{B(\mathbf{x})}(X) = \cup\{\mathbf{x}|B(\mathbf{x}) \subseteq \cup[B(\mathbf{x})|\mathbf{x} \in X]\}. \qquad (11)$$

The geometrical meaning of $\gamma(X)$ is the region of the space swept by all structuring sets $B(\mathbf{x})$ that are included in X. Extension to grey-level images can be tackled from different approaches [3,25]. Here we adopted for the bilateral case, a formulation based on binary processing the stack of cross-sections [19].

Examples. In Fig. 2 is provided a comparison of bilateral flat dilation and bilateral flat closing, i.e., $\delta_{B_{d,\Delta}(\mathbf{x})}(f)(\mathbf{x})$ and $\varphi_{B_{d,\Delta}(\mathbf{x})}(f)(\mathbf{x})$, with $d = 11$, of image $f(\mathbf{x})$ "Owl" with respect to different values of Δ, and using the morphological Gaussian kernel (i.e., parabolic) for the spatially-variant structuring



(a) Original

(b) Flat dil. 11 × 11

(c) Flat clos. 11 × 11

(b1) Bil. flat parab. dil. $d = 11$; $\varDelta = 5$

(b2) Bil. flat parab. dil. $d = 11$; $\varDelta = 15$

(b3) Bil. flat parab. dil. $d = 11$; $\varDelta = 45$

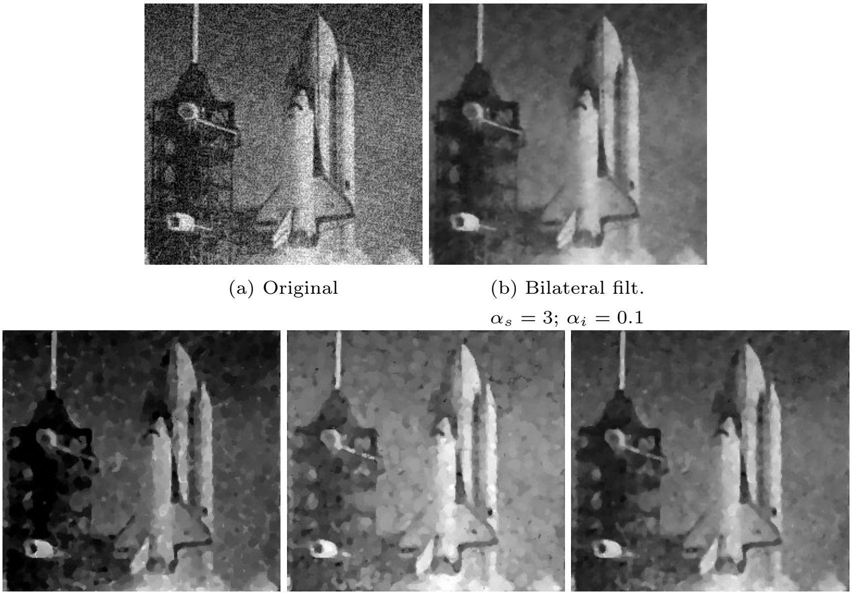
(c1) Bil. flat parab. clos. $d = 11$; $\varDelta = 5$

(c2) Bil. flat parab. clos. $d = 11$; $\varDelta = 15$

(c3) Bil. flat parab. clos. $d = 11$; $\varDelta = 45$

Fig. 2. Comparison of bilateral flat dilation $\delta_{B_{d,\varDelta}(\mathbf{x})}(f)(\mathbf{x})$ and bilateral flat closing $\varphi_{B_{d,\varDelta}(\mathbf{x})}(f)(\mathbf{x})$, with $d = 11$, of image $f(\mathbf{x})$ "Owl" with respect to different values of $\varDelta$, and using the morphological Gaussian kernel (i.e., parabolic) for the spatially variant structuring element. The results can also be compared with the standard spatially invariant flat dilation $\delta_B(f)(\mathbf{x})$ and flat closing $\varphi_B(f)(\mathbf{x})$.

element. It is observed in both cases the excellent adaptive behaviour of bilateral spatially-variant operators with respect to the standard ones. For $\varDelta = 5$, the main structural edges are preserved but an effect of dilation or closing is produced in the regular zones. In the case of $\varDelta = 15$, a stronger morphological filtering is observed, but the basic structures are still preserved. For high values of critical contrast percentage, e.g., $\varDelta = 45$, the results obtained are already quite similar to those of the spatially-invariant operators.



(a) Original

(b) Bilateral filt.
$\alpha_s = 3$; $\alpha_i = 0.1$

(d) Bil. flat parab. open. clos.
$d = 7$; $\varDelta = 5$

(e) Bil. flat parab. clos. open.
$d = 7$; $\varDelta = 5$

(f) Averaged (d) and (e)
$d = 7$; $\varDelta = 5$

Fig. 3. Comparison of image denoising using bilateral approaches: (a) original noisy image $f(\mathbf{x})$, (b) bilateral filtering (d) bilateral flat opening-closing (e) bilateral flat closing-opening (f) averaged image of opening-closing and closing-opening. The (logarithmic) Cauchy-Lorentz kernel has been used for the examples.

The last comparative example given in Fig. 3 deals with the problem of image denoising by means of bilateral approaches, using (Log.) Cauchy-Lorentz kernel. The latter kernel is a good compromise between the kernels of first (conical) and second order (parabolic). We observe that the standard bilateral filtering $\Upsilon(f)(\mathbf{x};\widetilde{k}_{\alpha_s,\alpha_i})$ produces a nice restoration. In the framework of mathematical morphology, the two products of an opening and a closing yield an interesting operator, the averaged alternate filter, which presents skilful properties for denoising. We remark in the example that the averaged alternate filter of diameter $d = 7$, using bilateral flat opening/closing, produces a filtered image which is also quite satisfactory. In particular, the restored edges with the latter filter are better enhanced than with the standard bilateral filtering of equivalent size. This property of enhancement of structural edges is fundamentally intrinsic to the flat morphological operators (based on max/min operations) with respect to the linear operators (based on averaging operation).

Complexity. The complexity of a direct implementation for the computation of the bilateral structuring functions is $O(W \cdot N)$ where W is the effective size of the spatial kernel and N is the total number of pixels of the image. Typically, for the studied kernels, one can consider only the pixels in a square window of size $2\eta_s \times 2\eta_s$, and hence $W \propto (2\eta_s)^2$. The complexity can be reduced to $O(\log W \cdot N)$ using a local histogram based algorithm [26]. In order to compare with previous approaches, and according to the recent paper [10], the complexity of the morphological amoebas is $O(N \cdot n\omega^2 \cdot \log \omega^2)$ and for the geodesic time-based neighbours is $O(N \cdot \omega^2 \cdot \log \omega^2)$, where n is the number of pixel graph connectivity and ω is the value of distance which defines the geodesic size. The computational advantage of bilateral structuring functions is natural since they are based on a Euclidean distance whereas the amoebas are founded in geodesic distances.

4 Conclusions and Perspectives

We have introduced in this paper the counterpart of bilateral filtering in the framework of mathematical morphology operators. Our starting point has been the notion of counter-harmonic filter, which provides a framework to explore the nonlinearization of convolution-based filters.

The main contributions of the paper can be summarised as follows. We have rediscovered the logarithmic connection between linear filters and the morphological ones, which allows us to propose the morphological equivalent to (locally adaptive) bilateral weighting convolution kernel: the bilateral structuring functions. We have considered, besides the Gaussian kernel, the use of a Laplace kernel or a Cauchy-Lorenz kernel for bilateral filtering and the corresponding logarithmic counterparts to define bilateral structuring functions. We have introduced a low-complexity framework for flat spatially-variant filters using thesholded adaptive structuring functions and a theoretical appropriate formulation of operators.

The performance of first experiments illustrated the applicative interest of the operators for adaptive nonlinear filtering. As mentioned above, a key point of the methods discussed in the paper is their low-complexity in comparison with other adaptive morphological frameworks.

From a theoretical viewpoint, the present work can be pursued into two main directions. On the one hand, using the slope transform [8,13], an investigation about the properties of the logarithmic Cauchy-Lorentz functions, with respect to the parabolic families which are better known, and in particular on the behaviour of the iteration of dilations/erosions using these non-standard functions can be achieved. On the other hand, the connection between the discrete operators and proposed here and the formulations of adaptive morphological operators [14,6] should be explored.

References

1. Angulo, J.: Pseudo-Morphological Image Diffusion using the Counter-Harmonic Paradigm. In: Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2010, Part I. LNCS, vol. 6474, pp. 426–437. Springer, Heidelberg (2010)

2. van den Boomgaard, R., Dorst, L.: The morphological equivalent of Gaussian scale-space. In: Proc. of Gaussian Scale-Space Theory, pp. 203–220. Kluwer, Dordrecht (1997)
3. Bouaynaya, N., Schonfeld, D.: Theoretical foundations of spatially-variant mathematical morphology part ii: Gray-level images. IEEE Trans. Pattern Anal. Mach. Intell. 30, 837–850 (2008)
4. Bullen, P.S.: Handbook of Means and Their Inequalities, 2nd edn. Springer, Heidelberg (1987)
5. Burgeth, B., Weickert, J.: An Explanation for the Logarithmic Connection between Linear and Morphological System Theory. International Journal of Computer Vision 64(2-3), 157–169 (2005)
6. Burgeth, B., Breuß, M., Pizarro, L., Weickert, J.: PDE-driven adaptive morphology for matrix fields. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 247–258. Springer, Heidelberg (2009)
7. Catte, F., Lions, P.-L., Morel, J.-M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM Journal on Numerical Analysis 29(1), 182–193 (1992)
8. Dorst, L., van den Boomgaard, R.: Morphological Signal Processing and the Slope Transform. Signal Processing 38, 79–98 (1994)
9. Florack, L., Maas, R., Niessen, W.: Pseudo-Linear Scale-Space Theory. International Journal of Computer Vision 31(2-3), 1–13 (1999)
10. Grazzini, J., Soille, P.: Edge-preserving smoothing using a similarity measure in adaptive geodesic neighbourhoods. Pattern Recognition 42(10), 2306–2316 (2009)
11. Heijmans, H., Ronse, C.: The Algebraic Basis of Mathematical Morphology. Part I: Dilations and Erosions. CVGIP: Image Understanding 50(3), 245–295 (1990)
12. Lerallut, R., Decenciere, E., Meyer, F.: Image Filtering Using Morphological Amoebas. Image Vision and Computing 25(4), 395–404 (2007)
13. Maragos, P.: Slope Transforms: Theory and Application to Nonlinear Signal Processing. IEEE Trans. on Signal Processing 43(4), 864–877 (1995)
14. Maragos, P., Vachier, C.: A PDE Formulation for Viscous Morphological Operators with Extensions to Intensity- Adaptive Operators. In: Proc. of ICIP 2008 (2008)
15. Maragos, P., Vachier, C.: Overview of adaptive morphology: trends and perspectives. In: Proc. of ICIP 2009 (2009)
16. Perona, P., Malik, J.: Scale-Space and Edge Detection Using Anisotropic Diffusion. IEEE Trans. Pattern Anal. Mach. Intell. 12(7), 629–639 (1990)
17. Paris, S., Kornprobst, P., Tumblin, J., Durand, F.: Bilateral Filtering: Theory and Applications. Foundations and Trends in Computer Graphics and Vision 4(1), 1–73 (2008)
18. Roerdink, J.: Adaptivity and group invariance in mathematical morphology. In: Proc. of ICIP 2009 (2009)
19. Serra, J.: Image Analysis and Mathematical Morphology. Theoretical Advances, vol. II. Academic Press, London (1988)
20. Soille, P.: Generalized geodesy via geodesic time. Pattern Recognition Letters 15(12), 1235–1240 (1994)
21. Soille, P.: Morphological Image Analysis. Springer, Berlin (1999)
22. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proc. of 6th Int. Conf. Computer Vision (ICCV 1998), New Delhi, India, pp. 839–846 (1998)
23. van Vliet, L.J.: Robust Local Max-Min Filters by Normalized Power-Weighted Filtering. In: Proc. of IEEE 17th International Conference of the Pattern Recognition (ICPR 2004), vol. 1, pp. 696–699 (2004)

24. Vachier, C., Meyer, F.: News from Viscousland. In: Proc. of ISMM 2007 (2007)
25. Verdú-Monedero, R., Angulo, J., Serra, J.: Anisotropic morphological filters with spatially-variant structuring elements based on image-dependent gradient fields. IEEE Trans. on Image Processing 20(1), 200–212 (2011)
26. Weiss, B.: Fast median and bilateral filtering. ACM Transactions on Graphics 25(3), 519–526 (2006)
27. Welk, M.: Families of generalised morphological scale spaces. In: Griffin, L.D., Lillholm, M. (eds.) Scale-Space 2003. LNCS, vol. 2695, pp. 770–784. Springer, Heidelberg (2003)
28. Welk, M., Breuß, M., Vogel, O.: Morphological amoebas are self-snakes. Journal of Mathematical Imaging and Vision 39(2), 87–99 (2011)

General Adaptive Neighborhood Viscous Mathematical Morphology

Johan Debayle and Jean-Charles Pinoli

Ecole Nationale Supérieure des Mines de Saint-Etienne
CIS-LPMG, UMR CNRS 5148
42023 Saint-Etienne Cedex 2, France

Abstract. This paper is focused on adaptive viscous morphology in the context of the General Adaptive Neighborhood Image Processing (GANIP) approach [1,2]. A local adaptive neighborhood is defined for each image point, depending on the intensity function of the image. These so-called General Adaptive Neighborhoods (GANs) are used as adaptive structuring elements for Mathematical Morphology (MM) [1,3]. In this paper, GAN-based viscous MM is introduced to adjust the filtering activity to the image gray levels. The proposed GAN viscous morphological filters are successfully applied on real application examples in image restoration and enhancement.

1 Introduction

1.1 The General Adaptive Neighborhood (GAN) Paradigm

This paper deals with 2D intensity images, that is to say image mappings defined on a spatial support D in the Euclidean space $\mathbb{R}^2$ and valued into a gray tone range, which is a real number interval. The General Adaptive Neighborhood paradigm has been introduced [4] in order to propose an original image representation for adaptive processing and analysis. The central idea is the notion of adaptivity which is simultaneously associated to the analyzing scales, the spatial structures and the intensity values of the image to be addressed.

In the so-called General Adaptive Neighborhood Image Processing (GANIP) approach [1,2], a set of General Adaptive Neighborhoods (GANs set) is identified around each point in the image to be analyzed. A GAN is a subset of the spatial support constituted by connected points whose measurement values, in relation to a selected criterion (such as luminance, contrast, thickness, ...), fit within a specified homogeneity tolerance. These GANs are used as adaptive windows for further image transformations or quantitative image analysis.

The space of *image* (resp. *criterion*) *mappings*, defined on the spatial support D and valued in a real number interval $\tilde{E}$ (resp. E), is represented in a General Linear Image Processing (GLIP) framework [5], denoted $\mathcal{I}$ (resp. $\mathcal{C}$). The GLIP framework $\mathcal{I}$ (resp. $\mathcal{C}$) is then supplied with an ordered vectorial structure, using

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 224–235, 2011.

the formal vector addition $\tilde{\oplus}$ (resp. $\oplus$), the formal scalar multiplication $\tilde{\otimes}$ (resp. $\otimes$) and the classical partial order relation $\geq$ defined as:

$$\forall (f,g) \in \mathcal{I}^2 \text{ or } \mathcal{C}^2 \quad f \geq g \Leftrightarrow (\forall x \in D \quad f(x) \geq g(x)) \tag{1}$$

The most frequently used framework is the *Classical Linear Image Processing* (CLIP) framework where the vector addition and the vector multiplication are the usual + and × operations, respectively. Additional GLIP frameworks have been developed for non-linear images or imaging systems, such as the *Logarithmic Image Processing* (LIP) framework [6].

1.2 GAN Sets

For each point $x \in D$ and for an image $f \in \mathcal{I}$, the GANs $V^h_{m_{\hexagon}}(x)$ are subsets in D. They are built upon a *criterion mapping* $h \in \mathcal{C}$ (based on a local measurement such as luminance, contrast, thickness, ...related to f), in relation with an *homogeneity tolerance* $m_{\hexagon}$ belonging to the positive intensity value range $E^{\oplus}$. More precisely, $V^h_{m_{\hexagon}}(x)$ is a subset of D which fulfills two conditions:

1. its points have a measurement value close to that of the point x : $\forall y \in V^h_{m_{\hexagon}}(x) \quad |h(y) \ominus h(x)|_{\hexagon} \leq m_{\hexagon}$, where $\ominus$ and $|.|_{\hexagon}$ denote the considered GLIP subtraction and GLIP modulus, respectively,
2. the set is path-connected (with the usual Euclidean topology on $D \subseteq \mathbb{R}^2$).

The weak GANs are thus defined as following:
$\forall (m_{\hexagon}, h, x) \in E^{\oplus} \times \mathcal{C} \times D$

$$V^h_{m_{\hexagon}}(x) = C_{h^{-1}([h(x) \ominus m_{\hexagon}, h(x) \oplus m_{\hexagon}])}(x) \tag{2}$$

where $C_X(x)$ denotes the path-connected component (with the usual Euclidean topology on $D \subseteq \mathbb{R}^2$) of $X \subseteq D$ containing $x \in D$ and $h^{-1}(Y) = \{x \in D; h(x) \in Y\}$ for $Y \subseteq E$.

The definition of $C_X(x)$ ensures that $x \in V^h_{m_{\hexagon}}(x)$ for all $x \in D$.

Figure 1 illustrates the GANs of two points computed with the luminance criterion in the CLIP framework on a human retina image. The figure highlights the homogeneity and the correspondence of the GANs with the spatial structures.

2 GAN Mathematical Morphology

In the litterature, several approaches have been investigated for defining adaptive morphological operators [7,1,8,9,10]. The basic idea in the General Adaptive Neighborhood Mathematical Morphology (GANMM) is to replace the usual Structuring Elements (SEs) by GANs, providing adaptive operators and filters [1,3].

2.1 Adaptive Structuring Elements

More precisely, the Adaptive Structuring Elements (ASEs), denoted $R^h_{m_{\hexagon}}(x)$, are defined as following:

$$\forall (m_{\hexagon}, h, x) \in E^{\oplus} \times \mathcal{C} \times D \quad R^h_{m_{\hexagon}}(x) = \bigcup_{z \in D} \{V^h_{m_{\hexagon}}(z) | x \in V^h_{m_{\hexagon}}(z)\} \tag{3}$$

The GANs $V^h_{m_{\hexagon}}(x)$ are not directly used as ASEs, because they do not satisfy the symmetry property contrary to the $R^h_{m_{\hexagon}}(x)$: $x \in R^h_{m_{\hexagon}}(y) \Leftrightarrow y \in R^h_{m_{\hexagon}}(x)$. This symmetry condition is relevant for visual, topological, morphological and practical reasons as explained in [4].

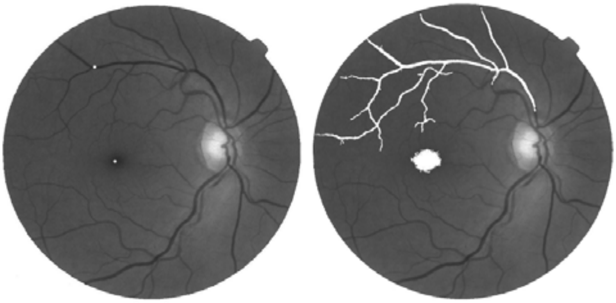
2.2 Adaptive Morphological Filters

The elementary operators of adaptive dilation and adaptive erosion are respectively defined accordingly to the ASEs:
$\forall (m_{\hexagon}, h, f) \in E^{\oplus} \times \mathcal{C} \times \mathcal{I}$

$$\mathrm{D}^h_{m_{\hexagon}}(f)(x) = \sup_{w \in R^h_{m_{\hexagon}}(x)} f(w) \tag{4}$$

$$\mathrm{E}^h_{m_{\hexagon}}(f)(x) = \inf_{w \in R^h_{m_{\hexagon}}(x)} f(w) \tag{5}$$

These two GAN-based operators, using adaptive structuring elements that are computed on the input or pilot image h, define input-adaptive morphological



(a) original image f with two seed points x and y (white dots)

(b) GANs $V^f_{10}(x)$ and $V^f_{10}(y)$

Fig. 1. The GANs of the two selected points of the original image (a) are homogeneous (b) with the tolerance $m = 10$, with respect to the luminance criterion in the CLIP framework

operators in the sense of [11]. So, they commute with suprema and infima and form an adjunction. The proofs are given below.

$$\begin{aligned}
\mathrm{D}^h_{m_{\hexagon}}(f \vee g)(x) &= \sup_{w \in R^h_{m_{\hexagon}}(x)} f \vee g(w) \\
&\Leftrightarrow \left(\sup_{w \in R^h_{m_{\hexagon}}(x)} f(w) \right) \vee \left(\sup_{w \in R^h_{m_{\hexagon}}(x)} g(w) \right) \\
&\Leftrightarrow \left(\mathrm{D}^h_{m_{\hexagon}}(f)(x) \right) \vee \left(\mathrm{D}^h_{m_{\hexagon}}(g)(x) \right)
\end{aligned}$$

The proof is similar for the adaptive erosion.

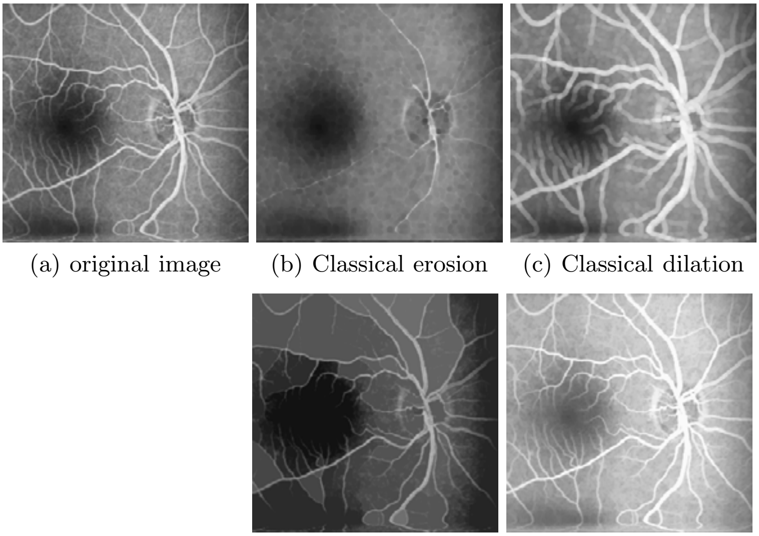
$$\begin{aligned}
\mathrm{D}^h_{m_{\hexagon}}(f) \leq g &\Leftrightarrow \mathrm{D}^h_{m_{\hexagon}}(f)(x) \leq g(x) \forall x \in D \\
&\Leftrightarrow \sup_{w \in R^h_{m_{\hexagon}}(x)} f(w) \leq g(x) \forall x \in D \\
&\Leftrightarrow f(w) \leq g(x) \forall w \in R^h_{m_{\hexagon}}(x), \forall x \in D \\
&\Leftrightarrow f(w) \leq g(x) \forall x \in R^h_{m_{\hexagon}}(w), \forall w \in D \\
&\Leftrightarrow f(w) \leq \inf_{x \in R^h_{m_{\hexagon}}(w)} g(x) \forall w \in D \\
&\Leftrightarrow f(w) \leq \mathrm{E}^h_{m_{\hexagon}}(g)(w) \forall w \in D \\
&\Leftrightarrow f \leq \mathrm{E}^h_{m_{\hexagon}}(g)
\end{aligned}$$

Using this input image h means that the ASEs in two successive runs will have the same shape, which results in the idempotence property in adaptive openings and closings.

Then, several adaptive morphological filters can be defined by combination of these two elementary adaptive morphological operators (opening, closing, alternating filters, sequential filters, alternating sequential filters ...) [3]. The GAN morphological operators satisfy several properties (increasing, adjunction, extensiveness, anti-extensiveness, commutativity with suprema or infima, idempotence, increasing with respect to $m_{\hexagon}$, translation invariance, multiplication compatibility) reported and proved in [1]. More particularly, in several and important practical cases, these adaptive operators are connected [1,3], which is an overwhelming advantage compared to the usual morphological ones that fail to this property [12]. For example, the GAN opening and closing are connected operators contrary to the classical opening and closing. Consequently, such GAN morphological operators perform an efficient spatially-adaptive image processing.

2.3 Classical vs. Adaptive Filtering

The GAN-based morphological filtering is more efficient than the classical morphological filtering. Indeed, an image is smoothed while both preserving the



(a) original image (b) Classical erosion (c) Classical dilation

(d) Adaptive erosion (e) Adaptive dilation

Fig. 2. Classical MM (using the centered disk of radius 3 as sliding window) vs. adaptive MM (using the GANs computed with the luminance criterion and the homogeneity tolerance value $m = 30$ in the CLIP framework)

transitions and the regions of interest. In particular, this kind of filters is very attractive for image restoration. A comparison of the classical and adaptive morphological filters is proposed in Figure 2 where the processing is performed on a human retinal image. The retinal vessels are rapidly damaged using the classical filtering contrary to the adaptive filters. In addition, no blurring effects occur with the GAN-based filtering.

3 GAN Viscous Mathematical Morphology

Gray tone MM could be defined from binary MM where the operator acts on the level sets of the image to be processed (stack filtering). Those stack filters can be extended to GAN-based viscous morphology, where the viscous notion enables to adjust the filtering activity to the image gray tones.

3.1 Stack Filtering

Stack filters, introduced in [13], are defined by applying a specific boolean operator to each threshold (level) of the gray tone image. They constitute a very important class of rank order based filters. They generalize classical filters such as (weighted) median filters, morphological filters ... [14]. The stack filtering principle is first briefly exposed before extending it within the GANIP framework.

A gray tone image $f : D \to \tilde{E}$ can be decomposed into its level sets X_t with $t \in \tilde{E}$ (the intensity value range of images):

$$X_t(f) = \{x \in D; f(x) \geq t\} \tag{6}$$

The gray tone image f can be naturally reconstructed from its level sets:

$$f = \bigvee_t t.\chi_t(f) \tag{7}$$

where χ_t is the indicator function of X_t, namely:

$$\chi_t(f)(x) = \begin{cases} 1 \text{ if } x \in X_t(f) \\ 0 \text{ otherwise} \end{cases}$$

Consider a binary operator ψ acting on the level sets associated to f, then the stack filtering consists in the following transformation denoted Ψ:

$$\Psi(f) = \bigvee_t t.\psi(\chi_t(f)), \quad \psi(\chi_t(f))(x) = \begin{cases} 1 \text{ if } x \in \psi(X_t(f)) \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

Such filters have to satisfy the stacking property: if the binary output images are piled on top of each other according their threshold level, the result for each point is always a column of 1's with a column of 0's on top. In this way, the output image could be easily and quickly reconstructed. Note that this property is satisfied when ψ is an increasing operator.

Of course, the filter ψ is just a Boolean kernel function operating within a sliding window. It enables to generalize a large class of filters satisfying the threshold decomposition architecture (i.e. filters commuting with any thresholding operation), such as median, max or min filters.

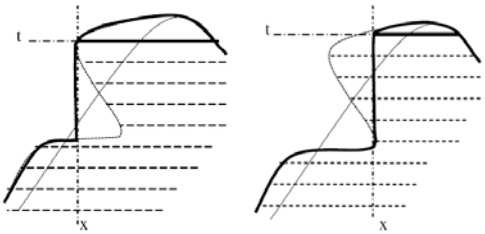
One can note that the classical filters process all level sets identically. On the contrary, viscous operators [15,16] process the level sets at different scales. It thus enables to define intensity-adaptive operators.

3.2 Classical Viscous Filters

A viscous filter is defined for a gray tone image f as:

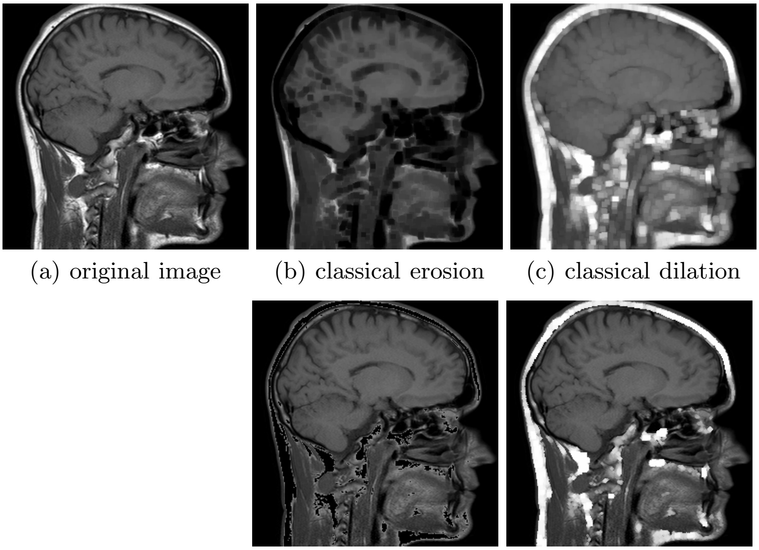
$$\Phi(f) = \bigvee_t t.\phi_t(\chi_t(f)) \tag{9}$$

where $\{\phi_t\}_t$, shortly denoted $\{\phi\}$ can be interpreted as a family of increasing set operators, each operator ϕ_t being assigned to act on the gray level t. If the family $\{\phi\}$ is decreasing, the reconstruction of the different filtered level sets is immediate, since the stacking property is satisfied. In the case of a non-decreasing family, the reconstruction is performed [10] by considering the lower or upper



(a) upper envelope (b) lower envelope

Fig. 3. Two stacking schemes for defining viscous filters [10]. Full line: initial stack; dotted line: filtered stack; bold line: final stack reconstruction. x denotes a position within the image spatial support and t a gray level.



(a) original image (b) classical erosion (c) classical dilation

(d) viscous erosion (e) viscous dilation

Fig. 4. Classical (b,c) vs viscous (d,e) erosion and dilation of the original image (a) using the function $t \mapsto r(t) = 3 * (t/256)^3$

envelope of the volume formed by the sets (Fig. 3). For example, the classical viscous dilation and viscous erosion are defined respectively as [16]:

$$\mathrm{D}_{\{r\}}(f) = \bigvee_t t.\mathrm{D}_{B_{r(t)}}(\chi_t(f)) \tag{10}$$

$$\mathrm{E}_{\{r\}}(f) = \bigvee_t t.\mathrm{E}_{B_{r(t)}}(\chi_t(f)) \tag{11}$$

where $\mathrm{D}_{B_{r(t)}}$ and $\mathrm{E}_{B_{r(t)}}$ denote a classical dilation and erosion, respectively, with a disk B of radius $r(t)$, a function of the gray level t.

Note that these two morphological operators do not define an adjunction in the general case [10].

As an example, Figure 4 shows an illustration of a classical erosion acting on a 8-bit image (with $M = 256$, the upper bound of the gray tone range) using a disk or radius 3 vs. a viscous erosion with the function $t \mapsto r(t) = 3 * (t/256)^3$. The upper envelope reconstruction is used since the family of viscous erosions $\{\mathrm{E}_{B_r(t)}\}_t$ is not decreasing.

Looking at the resulting images with the choice of the function r, points of highest luminance are strongly transformed while points of lowest luminance are left unchanged.

3.3 Adaptive Viscous Filters

Following the viscous notion, GAN-based viscous morphological filters can be defined in varying the homogeneity tolerance value $m_{\hexagon}$ of the adaptive dilation or erosion.

From a mathematical point of view, the adaptive viscous dilation and erosion are then defined respectively as:

$$\mathrm{D}^h_{\{m_{\hexagon}\}}(f) = \bigvee_t t.\mathrm{D}^h_{m_{\hexagon}(t)}(\chi_t(f)) \tag{12}$$

$$\mathrm{E}^h_{\{m_{\hexagon}\}}(f) = \bigvee_t t.\mathrm{E}^h_{m_{\hexagon}(t)}(\chi_t(f)) \tag{13}$$

where $m_{\hexagon}(t)$ is the varying homogeneity tolerance, a function of the gray level t.

These two GAN-based transforms are increasing operators ($\mathrm{D}^h_{m_{\hexagon}(t)}$ and $\mathrm{E}^h_{m_{\hexagon}(t)}$ are increasing for all t). Nevertheless, $(\mathrm{D}^h_{\{m_{\hexagon}\}}, \mathrm{E}^h_{\{m_{\hexagon}\}})$ does not define an adjunction in the general case. To get this morphological duality, it is necessary to introduce upper and lower transforms (Fig. 3) in the following way [10]:

$$\overline{\mathrm{D}}^h_{\{m_{\hexagon}\}}(f) = \bigvee_t t.\left(\bigvee_{s \geq t} \mathrm{D}^h_{m_{\hexagon}(s)}(\chi_s(f))\right) \tag{14}$$

$$\underline{\mathrm{D}}^h_{\{m_{\hexagon}\}}(f) = \bigvee_t t.\left(\bigwedge_{s \leq t} \mathrm{D}^h_{m_{\hexagon}(s)}(\chi_s(f))\right) \tag{15}$$

$$\overline{\mathrm{E}}^h_{\{m_{\hexagon}\}}(f) = \bigvee_t t.\left(\bigvee_{s \geq t} \mathrm{E}^h_{m_{\hexagon}(s)}(\chi_s(f))\right) \tag{16}$$

$$\underline{\mathrm{E}}^h_{\{m_{\hexagon}\}}(f) = \bigvee_t t.\left(\bigwedge_{s \leq t} \mathrm{E}^h_{m_{\hexagon}(s)}(\chi_s(f))\right) \tag{17}$$

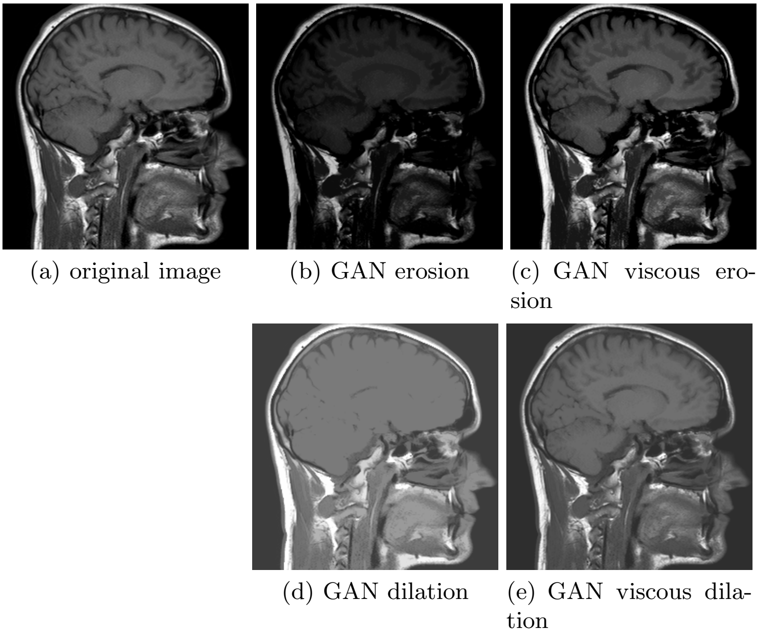
The operators $\overline{\mathrm{D}}^h_{\{m_{\hexagon}\}}$ and $\underline{\mathrm{D}}^h_{\{m_{\hexagon}\}}$ (resp. $\overline{\mathrm{E}}^h_{\{m_{\hexagon}\}}$ and $\underline{\mathrm{E}}^h_{\{m_{\hexagon}\}}$) coincide if and only if the family $t \mapsto \mathrm{D}^h_{m_{\hexagon}(t)}$ (resp. $t \mapsto \mathrm{E}^h_{m_{\hexagon}(t)}$) is decreasing. Since the couples

$(\mathrm{D}^h_{m_{\hexagon}(t)}, \mathrm{E}^h_{m_{\hexagon}(t)})$ define an adjunction for all t, the following products form a GAN-based viscous opening and closing, respectively:

$$\mathrm{O}^h_{\{m_{\hexagon}\}}(f) = \overline{\mathrm{D}}^h_{\{m_{\hexagon}\}} \circ \underline{\mathrm{E}}^h_{\{m_{\hexagon}\}}(f) \tag{18}$$

$$\mathrm{C}^h_{\{m_{\hexagon}\}}(f) = \underline{\mathrm{E}}^h_{\{m_{\hexagon}\}}(f) \circ \overline{\mathrm{D}}^h_{\{m_{\hexagon}\}} \tag{19}$$

As proposed in the classical case, an illustration (Fig. 5) of GAN-based viscous filtering is shown and compared with the GAN-based classical filtering. The filtering process is applied with the weak GANs using the luminance criterion $(h = f)$ within the CLIP framework. The homogeneity tolerance function for a 8-bit image $(M = 256)$ is defined as $t \mapsto m(t) = 30 * (1 - t/256)^3$. As previously mentionned, the results first show that images are not damaged by blur using these GAN-based filters. In addition, regarding the adaptive viscous filters, the image regions are naturally transformed in accordance with the decreasing function $t \mapsto m(t)$. Points with low luminance are largely transformed compared to those with high luminance. Therefore, the combination of these two characteristics (no blurring effects and controled filtering activity) provide efficient filters for image processing where structures of interest have a specific intensity range.



(a) original image (b) GAN erosion (c) GAN viscous erosion

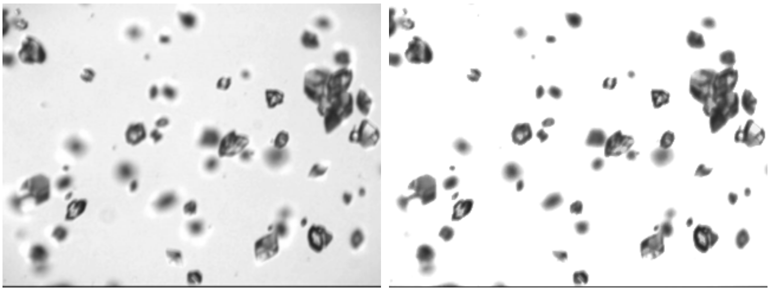
(d) GAN dilation (e) GAN viscous dilation

Fig. 5. GAN-based classical (b,d) vs viscous (c,e) dilation and erosion of the original image (a). The viscous filtering is performed with the function $t \mapsto m(t) = 30 * (1 - t/256)^3$.

4 Application Examples

4.1 Image Background Subtraction

The first application example concerns the processing of a 8-bit image ($M = 256$) coming from an in-situ cristallisation process of citric acids [17]. The specialists want to get both the size and shape distributions of the different cristals. Then, the cristals need to be segmented and characterized individually. For the segmentation step, a preprocessing is needed such as proposed in [17]. Indeed, the image background is not homogeneous. In the present paper, a background subtraction process (Fig. 6) is proposed by applying a GAN-based viscous morphological filtering using the function: $t \mapsto m(t) = 30 * (t/256)^3$. This function is used so that the filter mainly acts on image points of high intensity. The image points of low intensity are left unchanged.



(a) original image (b) GAN viscous dilation

Fig. 6. Background subtraction on a cristal image using the GAN-based viscous dilation

The result shows that the image background is well processed without damaging the size or shape of the cristals. Indeed, the filtering activity is specifically focused on image structures of high luminance, i.e. the background. In this way, the cristals could be more easily segmented and then characterized.

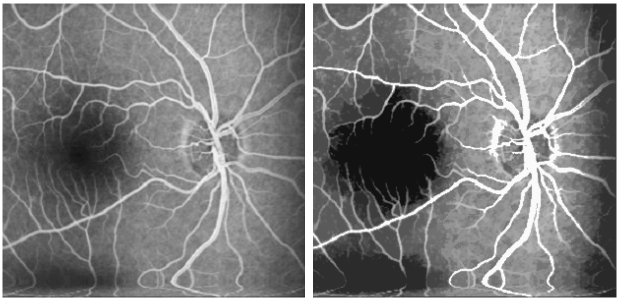
4.2 Image Enhancement

Another application of image contrast enhancement is also proposed in Figure 7. A 8-bit image ($M = 256$) of retinal vessels is filtered with a GAN-based viscous toggle filter using the function $t \mapsto m(t) = 90 * (t/256)^5$ in order to enhance the image. From a mathematical point of view, this toggle filter is defined as:

$$\kappa^h_{\{m_{\hexagon}\}}(f)(x) = \begin{cases} \mathrm{D}^h_{\{m_{\hexagon}\}}(f)(x) \text{ if } \mathrm{D}^h_{\{m_{\hexagon}\}}(f)(x) - f(x) < f(x) - \mathrm{E}^h_{\{\overline{m_{\hexagon}}\}}(f)(x) \\ \mathrm{E}^h_{\{\overline{m_{\hexagon}}\}}(f)(x) \text{ otherwise} \end{cases} \quad (20)$$

where $\overline{m_{\hexagon}}(t) = m_{\hexagon}(M - t)$.

This figure shows that retinal vessels (high luminance) become brighter and background regions (low luminance) become darker. So, the vessels are strongly



(a) original image (b) GAN viscous toggle filtering

Fig. 7. Image enhancement of human retinal vessels using the GAN-based viscous toggle filter

highlighted compared to the background. In addition, this filter enables to get more homogeneous vessels due to the filtering activity (homogeneity tolerance values). The resulting image is consequently enhanced and suitable for postprocessing.

5 Conclusion

In this paper, adaptive viscous morphology have been introduced in the context of the General Adaptive Neighborhood Image Processing (GANIP) approach. The resulting morphological filters show a high performance by processing an image while preserving its regions without damaging its transitions. The combination of GAN-based filtering and viscous morphology is particularly efficient in the sense that the filtering is adaptive to the image spatial structures and its activity is controled according to the image gray-tones. The theoretical advantages of these adaptive operators have been practically highlighted on real application examples for background subtraction of cristal images and enhancement of retinal vessel images. In addition, the computational cost of the GAN-based morphological filters is relatively low. Indeed, the computation time of the GAN-based morphological filters is about 20 seconds for a 256×256 image using a Pentium IV (3 GHz/2 GB RAM) and the Matlab software.

Currently, the authors are working on the Logarithmic Adaptive Neighborhood Image Processing (LANIP) [18] framework for viscous morphological filtering.

References

1. Debayle, J., Pinoli, J.C.: General Adaptive Neighborhood Image Processing - Part I: Introduction and Theoretical Aspects. Journal of Mathematical Imaging and Vision 25(2), 245–266 (2006)

2. Debayle, J., Pinoli, J.C.: General Adaptive Neighborhood Image Processing - Part II: Practical Application Examples. Journal of Mathematical Imaging and Vision 25(2), 267–284 (2006)
3. Pinoli, J.C., Debayle, J.: General adaptive neighborhood mathematical morphology. In: IEEE International Conference on Image Processing, Cairo, Egypt, November 7-10, pp. 2249–2252 (2009)
4. Debayle, J., Pinoli, J.C.: Spatially Adaptive Morphological Image Filtering using Intrinsic Structuring Elements. Image Analysis and Stereology 24(3), 145–158 (2005)
5. Pinoli, J.C.: A General Comparative Study of the Multiplicative Homomorphic, Log-Ratio and Logarithmic Image Processing Approaches. Signal Processing 58, 11–45 (1997)
6. Jourlin, M., Pinoli, J.C.: Logarithmic Image Processing: The Mathematical and Physical Framework for the Representation and Processing of Transmitted Images. Advances in Imaging and Electron Physics 115, 129–196 (2001)
7. Cuisenaire, O.: Locally adaptable mathematical morphology using distance transformations. Pattern Recognition 39(3), 405–416 (2006)
8. Lerallut, R., Decencière, E., Meyer, F.: Image filtering using morphological amoebas. Image and Vision Computing 25(4), 395–404 (2007)
9. Bouaynaya, N., Schonfeld, D.: Theoretical foundations of spatially-variant mathematical morphology. Part II: Gray-Level Images. IEEE Transcations on Pattern Analysis and Machine Intelligence 30(5), 837–850 (2008)
10. Vachier, C.: Upper and lower grey-level adaptive morphological operators. In: International Conference on Advances in Pattern Recognition, Kolkata, India, pp. 77–80 (2009)
11. Roerdink, J.B.T.M.: Adaptivity and Group Invariance in Mathematical Morphology. In: IEEE International Conference on Image Processing, Cairo, Egypt, November 7-10, pp. 2253–2256 (2009)
12. Salembier, P., Serra, J.: Flat Zones Filtering, Connected Operators, and Filters by Reconstruction. IEEE Transactions on Image Processing 4(8), 1153–1160 (1995)
13. Wendt, P.D., Coyle, E.J., Gallagher, N.C.: Stack filters. IEEE Transactions on Acoustics, Speech and Signal Processing 34(4), 898–911 (1986)
14. Hirata, N.S.T.: Stack filters: from definitions to design algorithms. In: Advances in Imaging and Electron Physics, vol. 152, pp. 1–47. Elsevier, Amsterdam (2008)
15. Vachier, C., Meyer, F.: The viscous watershed transform. Journal of Mathematical Imaging and Vision 22(2), 251–267 (2005)
16. Maragos, P., Vachier, C.: A pde formulation for viscous morphological operators with extensions to intensity-adaptive operators. In: IEEE International Conference on Image Processing, pp. 2200–2203 (2008)
17. Presles, B., Debayle, J., Fevotte, G., Pinoli, J.C.: A novel image analysis method for in-situ monitoring the particle size distribution of batch crystallisation process. Journal of Electronic Imaging 19(3), 1–7 (2010)
18. Pinoli, J.C., Debayle, J.: Logarithmic adaptive neighborhood image processing (lanip): Introduction, connections to human brightness perception and application issues. Journal on Advances in Signal Processing - Special issue on Image Perception 2007, 22 pages (2007) Article ID 36105

Spatially-Variant Structuring Elements Inspired by the Neurogeometry of the Visual Cortex

Miguel A. Luengo-Oroz

Biomedical Image Technologies, Universidad Politécnica de Madrid, Spain
maluengo@die.upm.es

Abstract. The V1-region of the primary visual cortex performs contour integration in the early mammalian visual system. The geometry of the neural connections of the V1-region has been mathematically described as a roto-translational continuous space. In this work, a bio-inspired methodology for processing 2D images based on the V1-region neurogeometrical structure is proposed. The input image is first transformed into the 3D roto-translational space. Then spatially-variant mathematical morphology operators using helicoidal structuring elements are applied in order to mimic the neural processing in the primary visual cortex. Finally, the output is projected back to the 2D cartesian space. Some illustrative straightforward applications of this methodology are presented for contour-completion and object-occlusion 2D problems.

1 Introduction

One of the most fascinating properties of the mammalian visual system is its capability for binding and segmenting incomplete patterns as illusory contours, i.e. the famous Kanizsa triangle (see Fig. 1). The primary visual cortex is said to play a fundamental role in the binding operation. Thus, replicating this behavior can be extremely useful for image analysis methods [1,2,3]. Other bio-inspired computer vision systems that mimic the first stages of the visual pathways - computer retinas- have been used for edge detection [4] or motion estimation [5]. Concerning the primary visual cortex, previous works of [6,7] have proposed mathematical continuous models for the neural architecture of this region. Computer simulations of neural activity using these models allow to extract illusory contours in the same way that vision does. Furthermore, image processing filtering techniques based on diffusion processes on this neural mathematical space have been proposed [8]. The present article is inspired by the previous mathematical models of the functional architecture of the V1-region in the primary visual cortex. The main idea is to propose a methodology that transforms a 2D image into the 3D roto-translational space and process the data in this 3D space using spatially-variant mathematical morphology. The neural connections are given by helicoidal structuring elements, and neuron synchronization - that produces the contour integration property - is replicated with a morphological filtering. This methodology can be used to effectively approach contour-completion or shape-occlusion image processing problems.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 236–247, 2011.

2 Background

2.1 The Primary Visual Cortex and Its Geometrical Structure

The visual pathway starts in the photoreceptors of the retina, a layer of cells at the back of the eye. The information leaves the eye by way of the optic nerve and arrives at the primary visual cortex, at the back of the brain. The receptive fields of the neurons in the early visual cortex (V1) characterize the response to different light patterns coming from the visual field. This response is frequently selective to oriented stimulus in a small region [9]. Horizontal connections between cells that join regions corresponding to receptive fields not overlapped in the visual field, are in charge of facilitating the contour integration. An important characteristic of the cortical design of V1 is the arrangement of its composing cells (neurons) into orientation columns. One orientation column corresponds to one point in the visual field and the coordinate along the column corresponds to the orientation angle (see Fig. 2). The role of this arrangement is to facilitate communication between cells that have similar orientation preferences but that are located in different visual field positions. There are several anatomical and neurophysiological studies in cats, monkeys, etc. that show the long range connections between cells of similar orientation preference. Furthermore, in [10], it has been shown that these connections extend for a longer distance along the "good continuation" and less in the orthogonal direction. This idea of perceptual binding has been previously described by the Gestalt's psychologists in the 50's [11] and has been formalized as the associative field [12](see Fig. 1), suggesting that a discontinued visual line will tend to be perceptually grouped together if the different segments are aligned or follow a smooth curve. In [6], a geometrical model of the column arrangement is proposed. The orientation columns are considered as vertical fibres distributed over a retinotopic array. Each point of the retinal field $r = (x, y) \in \Re^2$ is associated with the fibration $s = \frac{dx}{dy} \in S^1$. Thus, the geometrical space associated with the cell column arrangement is $(\Re^2, S^1)$, the roto-translational group [7]. In this organization, the arrangement of neighbouring fibers ensures all orientations in nearby columns are in a position to interact between them. When a signal arrives at the retinotopic field, it passes to the visual cortex and the signal is converted into the V1-column model: $\Re^2 \rightarrow (\Re^2, S^1)$. Afterwards, each cell acts as a single processor that interacts with its cell neighborhood regarding its geometrical context.

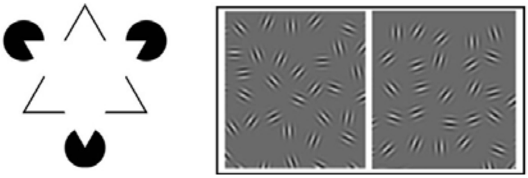


Fig. 1. (left) Kanizsa's triangle. (right) Although both images have similar statistic properties, we "see" a continuous line on the left image.

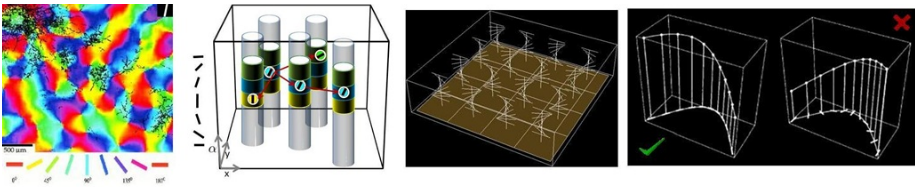


Fig. 2. From left to right. (1)Relationship between orientation columns and long range horizontal connections revealed in optical imaging of tree shrew V1 region combined with injection of biocytin tracer. Different colors represent the different orientation columns. The white dots represent the region where an anatomical tracer is injected. The black dots represent axonal terminals colored by this process. Long range interactions tend to concentrate in cortical regions that share similar orientation specificity with the injection site. Shorter range interactions span other orientations, the more orthogonal with the initial orientation, the shorter the interaction [extracted from [10]]. (2) Schematic representation of column organization with longer horizontal inter-columnar connections between iso-orientation cells. (3) Arrangement of preferred orientations for columns in the $(\Re^2, S^1)$ space. Cells in a plane with $s = K$ have the same preferred orientation. (4) Not all the arbitrary paths in the 3D volume of $(\Re^2, S^1)$ correspond to a continuous line in the retinotopic field $(\Re^2)$ [adapted from [6]].

When neurons process signals in such a space, contour integration based on the concept of associative field emerges in a simple natural way.

2.2 Mathematical Morphology Basics

In the framework of digital grids, a *binary image* can be represented by a function $f : D_f \rightarrow T$, where D_f is a subset of $\mathbf{Z}^2$ and $T = \{0, 1\}$ is an ordered set. Let B be a subset of $\mathbf{Z}^2$ and $\lambda \in \mathbf{N}$ a scaling factor. λB is called *structuring element* (shape probe) B of size λ. The basic morphological operators are dilation $(\delta_B(f(x)) = \sup_{y \in B}\{f(x-y)\})$ and erosion $(\varepsilon_B(f(x)) = \inf_{-y \in B}\{f(x-y)\})$. A dilation by a structuring element λB can by obtained by iteration of the unit structuring element B, hence $\delta_{\lambda B}(f) = \delta_{(\lambda-1)B}\delta_B(f)$ The two elementary operations of *erosion* and *dilation* composed together give opening $(\gamma_{\lambda B}(f) = \delta_{\lambda B}[\varepsilon_{\lambda B}(f)])$ and closing $(\varphi_{\lambda B}(f) = \varepsilon_{\lambda B}[\delta_{\lambda B}(f)])$, that filter out light and dark structures from the images according to a predefined size λ and shape criterion B. Although image processing applications using morphological operators commonly use euclidean-based structuring elements (disk or sphere), recent works propose using different structuring elements in the same image, depending on its contextual or geometric information [13,14,15,16]. In [17], we find a very interesting work that deals with adaptive morphology for edge linking. Generally speaking, the structuring element fixes the metric in the image space.

3 Mimicking Neural Processing with Morphological Operators in the Neurogeometrical Space

0The processing chain of our model works as follows (see Fig. 3): first the input image is lifted into the roto-translational space. Then morphologicaloperators

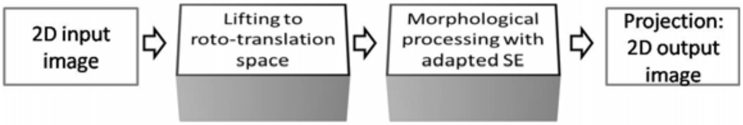

2D input image
Lifting to roto-translation space
Morphological processing with adapted SE
Projection: 2D output image

Fig. 3. Processing workflow

mimicking the neural cell-to-cell interactions are performed according to position-dependant structuring elements. The structuring elements are embedded in the $(\mathbb{Z}^2, S^1)$ space. Finally the image is projected back to the image space. This processing can be seen as a filtering process and its properties depend on the morphological operators performed in the roto-translational space.

3.1 Lifting Up Images: From $(\mathbb{Z}^2)$ to $(\mathbb{Z}^2, S^1)$

In the visual system, the signal acquired by photoreceptors in the retina is transformed into an active pattern in the neural hypercolumn structure of the visual cortex. Similarly, in our system, the signal passes from the image space $f \in (\mathbb{Z}^2)$ to the roto-translational space $f' \in (\mathbb{Z}^2, S^1)$. This procedure can easily be done by means of morphological linear openings: $f'(x, y, s) = \gamma_{B(s)}(f(x, y))$, where $B(s)$ corresponds to a line structuring element with a rotation angle of s (see Fig. 4). Note that the process of lifting is inherently imperfect (due to the discretization of the angle) and its precision should be carefully evaluated in future research [18]. For simplicity, in the examples showed in this article, a nearest neighbor approximation has been used to calculate the segments rotated at different angles.

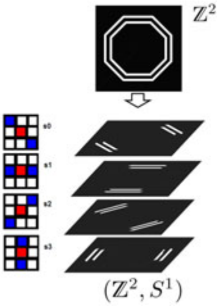


Fig. 4. A simple discrete configuration of $(\mathbb{Z}^2, S^1)$ is illustrated. The S^1 space is decomposed into 4 possible orientations, $\{s0, s1, s2, s3\}$, associated respectively to a rotation angle of $\{-45, 0, 45, 90\}$ degrees. Lifted image is $f'(x, y, s) = \gamma_{B(s)}(f(x, y))$, $s = \{s0, s1, s2, s3\}$. On the left, the structuring elements $B(s)$ associated to each level of S^1.

3.2 Structuring Elements in $(\mathbb{Z}^2, S^1)$

The structuring element (SE) associated with a position in the visual field $r = (x, y)$ and an orientation s models the connections of the neuron at position $p = (r, s) \in (\mathbb{Z}^2, S^1)$. These connections represent the possibilities allowed for a "good continuation" from p (see Fig. 1). Thus, in each level corresponding to an angle s, there exists a privileged direction associated with the horizontal connections (see Fig. 2). Two points $a(r, s)$ and $b(r', s)$ can be connected if r and r' are in the same straight line of orientation s. Two points $a(r, s)$ and $b(r', s + \Delta s)$, that are not collinear, can be connected if the admitted "good continuation" allows a turn of at least Δs. Consequently, the SE associated with the position p is a partial helicoid whose rotation axis is perpendicular to the plane $\mathbb{Z}^2$. The helicoid rotation angle and its radius are fixed by the turn admitted in the associative field. Globally, the connectivity of the $(\mathbb{Z}^2, S^1)$ space (equivalent to the neural wiring) is given by an SE which is position-invariant for the SE's defined over a plane $s = K$. Let $St_{(r,s)}$ be the SE in the position $p(r, s)$; so $\forall r' \in \mathbb{Z}^2, St_{(r',s)} = St_{(r,s)}$. The SE associated with an orientation s is denoted St_s. As a result of the arrangement of orientation columns, $St_{s+\Delta s} = St_s$ *rotated* Δs. Therefore the SE is position-dependent and it turns when changing the s coordinate (see Fig. 5). Note also that the support of the $(\mathbb{Z}^2, S^1)$ space is periodic along the s coordinate (the planes $s = min$ and $s = max$ are neighbors). All of the paths in the $(\mathbb{Z}^2, S^1)$ space produced by the lift of a 2D image whose content is a continuous connected component, only correspond to a subset of all the possible connected paths with euclidian metric in the 3D space of $(\mathbb{Z}^2, S^1)$. Minimal paths, calculated on the distance function obtained with the metric fixed by the proposed SE's, are equivalent to geodesics in the $(\Re^2, S^1)$ space.

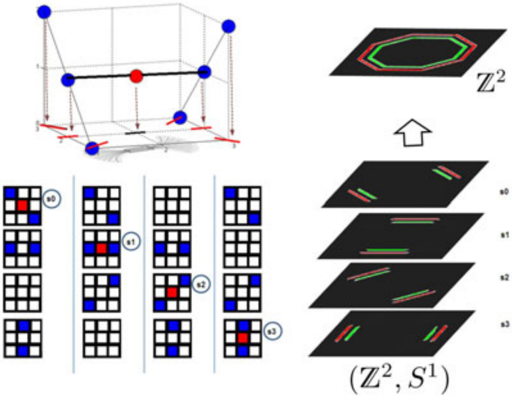


Fig. 5. (top-left) Example of structuring element associated to the space $(\mathbb{Z}^2, S^1)$,$S^1 = \{s0, s1, s2, s3\}$ and its projection on the associative field. The "good continuation" admits a turn of 45 degrees. (bottom-left) The specific SE's in function of the plane $\{s0, s1, s2, s3\}$ where they are located; in red the central point and in blue its neighbors. (right) Labeling of connected elements using the proposed SE's on the lifted image displayed in the previous figure. Results are projected back to $\mathbb{Z}^2$.

3.3 Morphological Neural Processing

The visual cortex is represented by the lattice points and its functionality modeled with a type of simple neural network. Instead of starting from the classical approach where the state of each neuron is calculated in function of its entries, i.e. $a_i(t+1) = f(\sum_j a_j(t)w_{ji} + \theta_i)$, we chose a dual approach: what is the influence of one neuron on its neighborhood? In the simplest binary mode a neuron propagates an excitatory signal by turning on all the neurons that are connected to it (we denote B_i neurons connected with the axon of neuron i). By analogy, a neuron propagates an inhibitory signal to its neighborhood by turning them off. So the equations that describe the activity $\{0,1\}$ of a neuron a_i over time are: $a_i(t+1) = \bigvee_{j/i \in B_j} a_j(t)$, and $a_i(t+1) = \bigwedge_{j/i \in B_j} a_j(t)$. The $\bigvee$ (max) equation describes the behavior of an excitatory signal propagation and the $\bigwedge$ (min) equation the inhibitory signal propagation. Each neuron is a pixel and ,in morphological terms, it is connected to its neighbors that correspond to the structuring element associated with that pixel. Excitation and inhibition are also interpreted in terms of morphology as dilation and erosion (see Fig. 6). Therefore the structuring elements define the neural hardware of our system. There is a particularly simple case in these terms,when the cable (ie. a neural fiber) between two cells is bidirectional and therefore cells affected by cell i all affect cell i. These kinds of systems are associated with the symmetric and invariant with respect to translation structuring elements, including segments, squares and hexagons. The simplified equations for a system with these structuring elements are: $a_i(t+1) = \bigvee_{j \in B_i} a_j(t)$, and $a_i(t+1) = \bigwedge_{j \in B_i} a_j(t)$. Therefore, neural wires are completely defined by the structuring elements associated to each position. The evaluation function of each neuron is simplified to a max/min operation, in morphological terms, dilation and erosion (note the iterative and synchronous character of the proposed model is a simplified appoproach, opposed to the asyncronous firing and synchronization of biological neurons in the brain). The mathematical function proposed to perform the contour integration is a closing. A morphological closing is composed by a dilation and its dual erosion ($\varphi_B(f) = \varepsilon_B[\delta_B(f)]$). If the image represents a topographic surface, a morphological closing φ fills channels and lakes and completes partially gulfs. If we interpret the closing sequentially over time in the visual cortex lattice, the dilation models the propagation of an excitatory signal from a single neuron to its neighbors. In case two near neurons are simultaneously activated, a nexus between those two neurons will be created - the perceptive illusion of a line integrated between two points. When an isolated neuron is activated, the sequence of dilations and erosions (=closing) will cause

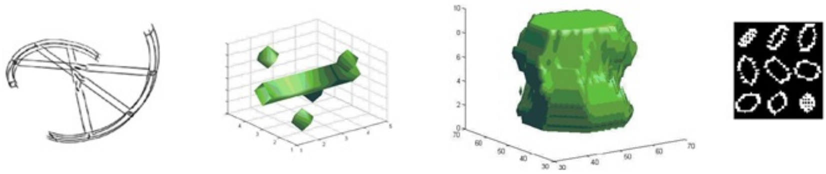


Fig. 6. From left to right: (1) Schematic representation of helicoidal structuring element (similar to DNA double helix organization). (2) Discrete SE with collinear size of 5 pixels. (3) Volume rendering and (4) cross-sections of the wavefront resulting in five iterative dilations of the SE by itself.

the neuron to return to its initial state. The output of the closing operator can be considered as the stable response of the propagation of an excitatory signal (See Fig. 7). Properties of closings can be interpreted as a high level behavior of visual perception. *Increasing*: when vision performs the contour integration, it creates contours but it does not make them disappear. *Idempotent*: regardless of how long we look at an object, this is not going to cause it to change its shape. *Translation invariance*: this property is generally lost in the $(\mathbb{Z}^2, S^1)$ space defined with helicoidal structuring elements, as happens usually in the spatially-variant morphological framework. However, when the translations are made in the $(\mathbb{Z}^2)$ space before lifting up the image to $(\mathbb{Z}^2, S^1)$, the output of the closing operator remains invariant. That means that the translation invariance works only for the subset of translations in $(\mathbb{Z}^2, S^1)$ generated by lifting up objects moved in $(\mathbb{Z}^2)$, corresponding to objects moving in the visual field.

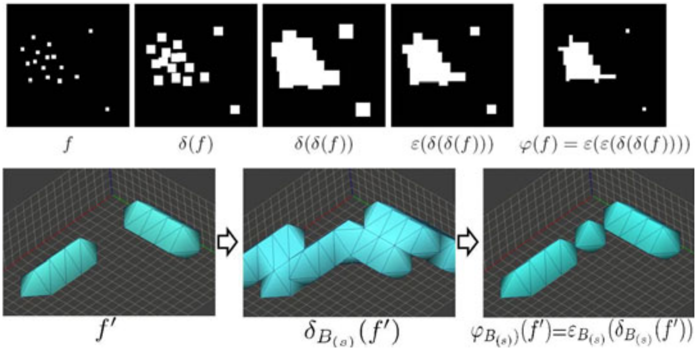

f
$\delta(f)$
$\delta(\delta(f))$
$\varepsilon(\delta(\delta(f)))$
$\varphi(f) = \varepsilon(\varepsilon(\delta(\delta(f))))$
f'
$\delta_{B_{(s)}}(f')$
$\varphi_{B_{(s)})}(f')=\varepsilon_{B_{(s)}}(\delta_{B_{(s)}}(f'))$

Fig. 7. (top) Illustration of closing properties in a 2D example. Neurons which are next to each other, "synchronize" and fill the gap between themselves according with the utilized structuring element (a square in this example). (bottom) Closing example in $(\mathbb{Z}^2, S^1)$ space, where $S^1 = \{s0, s1, s2, s3\}$ and the structuring elements are those of Fig. 5. A pixel binding the gap between two objects with different orientations is added after closing.

3.4 Projecting Back to the Image Space $(\mathbb{Z}^2)$

In order to come back to the initial image space after performing the morphological closing in the $(\mathbb{Z}^2, S^1)$ space, all the planes at different s levels should be combined, i.e. $f''(x,y) = \bigvee_{\forall s \in S} f'(x,y,s)$. However, for the sake of analysis purposes, we may be interested in maintaining the roto-translation representation. For instance, it would allow us to have two different connected components for two orthogonal crossing lines. In fact, it is this kind of representation that is provided by the V1-region to other brain regions.

4 Application Examples

Hereafter, we present three examples that illustrate the potential applications of transforming and processing 2D images in the roto-translational space. The

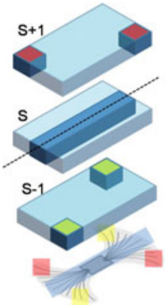


Fig. 8. Example of basic helicoidal SE in a plane S=s. The preferred orientation in this plane is fixed by the dotted axis. The SE is characterized by its size along the preferred (collinear) orientation and the Δs corresponding to the differences of favored orientations between two consecutive planes s and $s+1$. Scaled SE can by obtained by dilating the basic helicoid SE by itself. A projection of the pixels at different orientations belonging to this SE is shown in the lowest plane.

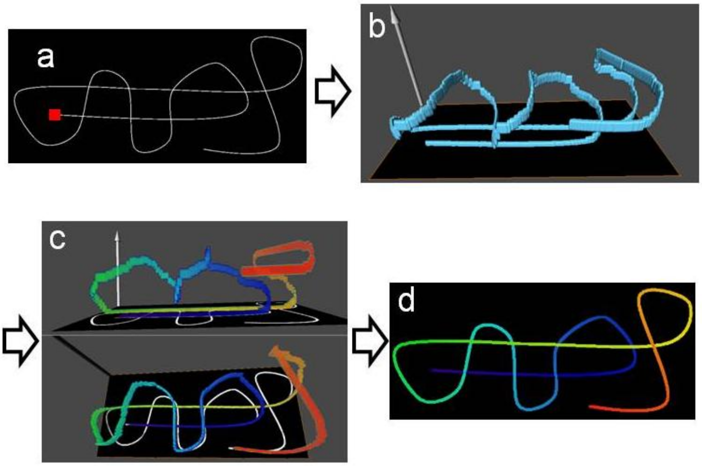


Fig. 9. Distance calculation on tangled line. *(a)* Input image; the red box is the beginning of the line. *(b)* Input image lifted into the roto-translation space. Image size is $(348, 771, 90)$, the size of the line structuring element (SE) used to lift the image is 11 pixels and ΔS between the preferred orientations of two planes at s and $s+1$ is $180/90 = 2$ degrees. *(c)* Geodesic distance on (b), starting from the red-box location (two different 3D views with the 2D projection in the lowest plane). Color code goes from dark blue (dist$= 0$) to red (dist$= max$). *(d)* Projection of (c) into the 2D image space.

unit helicoid SE used in the experiments is explained in Fig. 8. In the first example (Fig. 9), the geodesic distance function in $(\mathbb{Z}^2, S^1)$ is used to measure the length of a continuous tangled line in 2D. Distance function on the crossroads is correctly computed when the angle formed by the two crossing lines is

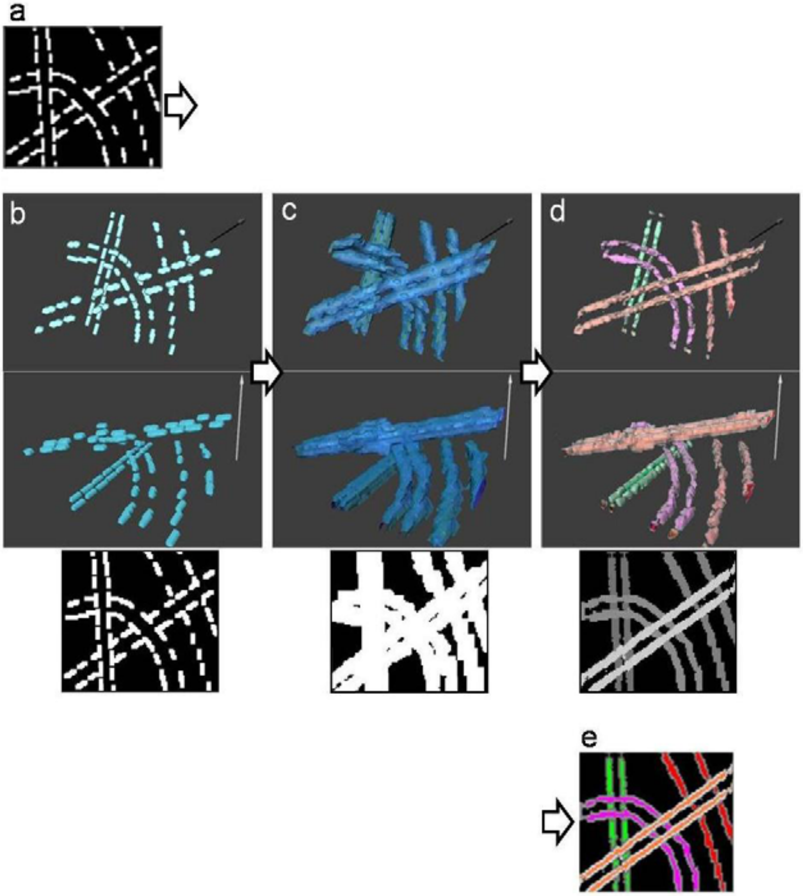


Fig. 10. Perceptual binding of crossing dotted lines. *(a)* Input image. *(b)* Input image lifted into the roto-translation space (volume renderings and projection). Image size is $(98, 100, 45)$, the size of the line SE used to lift the image is 5 pixels and ΔS between the preferred orientations of two planes at s and $s + 1$ is $180/45 = 4$ degrees (2 different 3D views of the image). *(c)* Dilation of (b) by a helicoidal SE formed by the dilation of a basic SE of collinear size of 7 pixels by itself. *(d)* Erosion of (c), therefore $(d) = \varphi(b) = \varepsilon[\delta(b)]$, and labeling of connected components (parallel lines were assigned the same label value). Note that all the original dotted lines are now separated connected components. *(e)* Projection of (d) into the 2D image space (only one color is showed in line crossings).

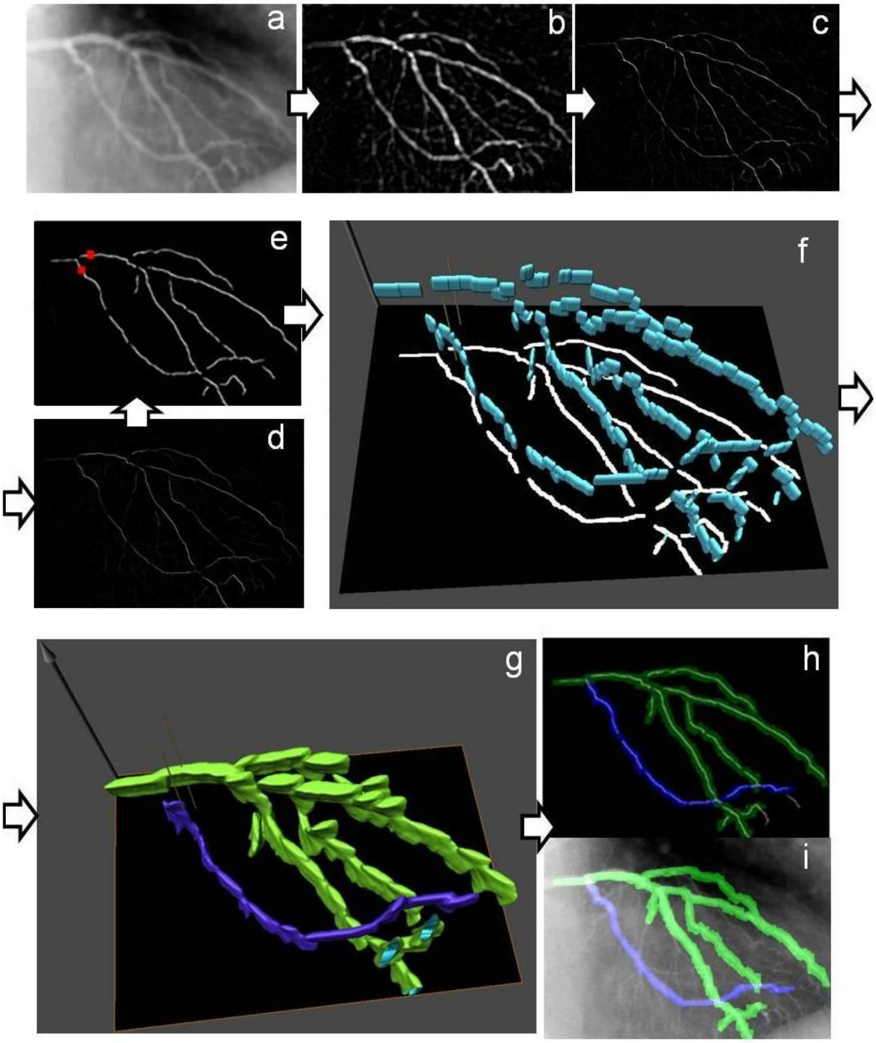


Fig. 11. Main arteries extraction on coronary angiography. *(a)* Input image. *(b)* Top-hat filtering of (a). *(c)* Grey tone homotopic thinning of (b). *(d)* Area opening of (c) . *(e)* Threshold on (d); red-boxes are markers for branch reconstruction. *(f)* Image (e) lifted into the roto-translation space; markers are represented as small thin vertical cilinders. Image size is $(400, 570, 90)$, the size of the line SE used to lift the image is 7 pixels and ΔS between the preferred orientations of two planes at s and $s + 1$ is $180/90 = 2$ degrees. *(g)* Geodesic morphological reconstruction of (f) from markers, (f) has been previously dilated two times by a basic helicoidal SE with a collinear size of 14 pixels. *(h,i)* Reconstructed branches projected to image space and superposed to (e) and (a).

bigger than the turning allowed by the helicoidal structuring element, that is, the connectivity rule in $(\mathbb{Z}^2, S^1)$. The second (Fig. 10) illustrates with a synthetic image how contour integration and disambiguation of occluded objects can be approached by the closing operator in the roto-translation space. Object labeling directly in $(\mathbb{Z}^2, S^1)$ allows to easily separate superposed objects in $(\mathbb{Z}^2)$. The last (Fig. 11) shows a short processing pipeline that identifies the two main vessel trees in a coronary angiography using a reconstruction by dilation operator. Note that, in the presented examples, several SE's corresponding to consecutive orientations are effetively the same due to the discretization of the angle using a nearest neighbour approximation. That means that one pixel at a certain positions migth have been lifted in several consecutive orientation planes of the space $(\mathbb{Z}^2, S^1)$. The question of how to use segments rotated by small angles in a discrete grid is limeted by the segment length and should be deeply explored by means of subpixel accuracy methods.

5 Discussion

We have presented an original bio-inspired image processing methodology that mimics the functions of the early mammalian visual system by transforming a 2D image into a 3D roto-translational space and applying morphological operators in this space. The neural wiring architecture of the V1-area of the visual cortex is replicated by helicoidal spatially-dependent structuring elements and the contour integration property is performed as a morphological filtering. The proposed methodology allows to efficiently approach contour-completion and object-occlusion image processing problems with very reduced mathematical machinery. Several potential applications of this framework can be easily envisaged, i.e. road tracking from satellite imaging or vessel extraction from medical images acquired with projective techniques (i.e. X-ray). The goal of this article is to propose and illustrate the use morphological operators in the roto-translational space. However, future developments should deeply explore the use of more complex morphological operators, different curvature-adapted helicoidal structuring elements and the expansion for grey-level image processing. Overall, this methodology can be used not only for mimicking the binding property of the visual system, but it can also provide a bio-inspired framework for approaching other image analysis problems.

Acknowledgements

This work was partially supported by LA CAIXA grant, FPI and ARTEMIS (Comunidad de Madrid, Spain) and the European Regional Development Funds.

References

1. Kokkinos, I., Deriche, R., Faugeras, O., Maragos, P.: Computational analysis and learning for a biologically motivated model of boundary detection. Neurocomputing 71 (2008)

2. Williams, L.R., Jacobs, D.W.: Stochastic completion fields: A neural model of illusory contour shape and salience. Neural Computation 9(4), 837–858 (1997)
3. Gavet, Y., Pinoli, J.C.: Visual perception based automatic recognition of cell mosaics in human corneal endothelium microscopy images. Image Analysis and Stereology 27, 53–61 (2008)
4. Mertoguno, S., Bourbakis, N.G.: A digital retina-like low-level vision processor. IEEE Transactions on Systems, Man, and Cybernetics, Part B 33(5), 782–788 (2003)
5. Barranco, F., Diaz, J., Ros, E., del Pino, B.: Visual system based on artificial retina for motion detection. IEEE Transactions on Systems, Man, and Cybernetics, Part B 39, 752–762 (2009)
6. Petitot, J., Yannick, T.: Vers une neurogéométrie. Fibrations corticales, structures de contact et contours subjectifs modaux. Mathématiques et sciences humaines 145, 5–101 (1999)
7. Citti, G., Sarti, A.: A cortical based model of perceptual completion in the roto-translation space. Journal of Mathematical Imaging and Vision 24(3), 307 (2006)
8. Sanguinetti, G., Citti, G., Sarti, A.: Image Completion Using a Diffusion Driven Mean Curvature Flowin A Sub-Riemannian Space. In: VISAPP (2), pp. 46–53 (2008)
9. Hubel, D.H., Wiesel, T.N.: Functional architecture of macaque monkey cortex. Proceedings of the Royal Society of London 198, 1–59 (1977)
10. Bosking, W.H., Zhang, Y., Schofield, B., Fitzpatrick, D.: Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex. Journal of Neuroscience 17(6), 2112–2127 (1997)
11. Kanizsa, G.: Organization in vision: Essays on gestalt perception (1979)
12. Field, D.J., Hayes, A., Heiss, R.F.: Contour integration by the human visual system: Evidence for a local" association field.".. Vision Research 33(2), 173–193 (1993)
13. Debayle, J., Pinoli, J.C.: Spatially adaptive morphological image filtering using intrinsic structuring elements. Image Analysis and Stereology 24, 145–158 (2005)
14. Bouaynaya, N., Charif-Chefchaouni, M., Schonfeld, D.: Theoretical Foundations of Spatially-Variant Mathematical Morphology . IEEE Transactions on Pattern Analysis and Machine Intelligence 30(5), 823–836 (2008)
15. Luengo-Oroz, M.A., Angulo, J.: Cyclic Mathematical Morphology in Polar-Logarithmic Representation. IEEE Transactions on Image Processing 18(5) (2009)
16. Verdú-Monedero, R., Angulo, J.: Spatially-Variant Directional Mathematical Morphology Operators Based on a Diffused Average Squared Gradient Field. In: Proceedings of the 10th ICACIVS, pp. 542–553. Springer, Heidelberg (2008)
17. Shih, F.Y., Cheng, S.: Adaptive mathematical morphology for edge linking. Information Sciences 167(1-4), 9–21 (2004)
18. Hendriks, C., Van Vliet, L.: Using line segments as structuring elements for sampling-invariant measurements. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1826–1831 (2005)

Towards a Parallel Topological Watershed: First Results

Joël van Neerbos[1], Laurent Najman[2], and Michael H.F. Wilkinson[1]

[1]Johann Bernoulli Institute, University of Groningen
joelvneerbos@gmail.com, m.h.f.wilkinson@rug.nl
[2]Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, A3SI, ESIEE
l.najman@esiee.fr

Abstract. In this paper we present a parallel algorithm for the topological watershed, suitable for a shared memory parallel architecture. On a 24-core machine an average speed-up of about 11 was obtained. The method opens up possibilities for segmentation of gigapixel images such as found in remote sensing routinely.

1 Introduction

The watershed transformation is a popular tool for segmenting grey-scale images, introduced by S. Beucher and C. Lantuéjoul [2]. It can be used to segment an image into regions with similar grey values. Due to ever increasing image sizes, several parallel algorithms have become available for different watershed paradigms [9]. In contrast, no parallel algorithms for the *topological watershed* [1,3] are available. In the topological watershed framework, some of the grey-scale information from the original image is preserved, which may be useful for further processing, such as reconnection of corrupted contours. Also, this grey-scale information can be used to determine the significance of watershed lines [7].

In this paper we present a parallel algorithm for the topological watershed on shared-memory parallel machines. The algorithm is based on parallelizing each of the stages of the sequential algorithm [4], and including a multi-pass stage such as used in most parallel algorithms for the regular watershed [9].

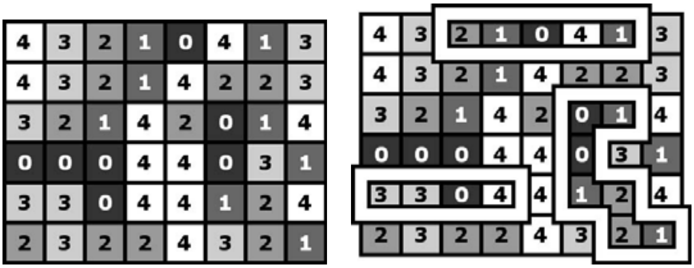
We will first describe the topological watershed briefly, after which we describe the parallel algorithm. Timing results are presented in Section 4, followed by the conclusions.

2 Topological Watersheds

The topological watershed [1,3] was introduced to include grey level information in the end result, in such a way that the significance of each watershed line is preserved. Specifically, the topological watershed preserves the *pass values* [1,8], i.e. the highest altitude of the lowest path between any two minima.

The pass value of two points in the image is related to a concept of *separation* of the points. The points p and q are said to be *k-separated* if the following conditions apply:

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 248–259, 2011.



4	3	2	1	0	4	1	3
4	3	2	1	4	2	2	3
3	2	1	4	2	0	1	4
0	0	0	4	4	0	3	1
3	3	0	4	4	1	2	4
2	3	2	2	4	3	2	1

(a)

4	3	2	1	0	4	1	3
4	3	2	1	4	2	2	3
3	2	1	4	2	0	1	4
0	0	0	4	4	0	3	1
3	3	0	4	4	1	2	4
2	3	2	2	4	3	2	1

(b)

Fig. 1. Separation. (a) a digital grey-scale image; (b) lowest paths between three pairs of pixels.

- There exists a path from p to q with maximum value $k - 1$
- There exists no path from p to q with a maximum value lower than $k - 1$
- Both p and q have a value lower than $k - 1$

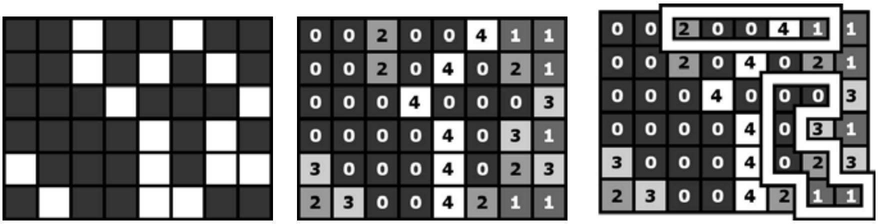
A path that satisfies the first two conditions for some k is called a *lowest path* in this paper. If a lowest path from p to q contains no value that is higher than both p and q, then p and q are not separated, but are *linked*. Separation is illustrated in Fig. 1. The top path in Fig. 1(b) connects two 5-separated pixels. The left path connects pixels that are not separated (but linked). The right path connects two 3-separated pixels.

In contrast to most watershed algorithms, topological watershed does not produce a binary image, but a grey-scale image. Intuitively, grey values of the pixels are obtained as follows:

- All pixels in a basin have the same grey value, namely the value of the minimum from the input image that is contained within the basin.
- The values of the pixels on the watershed lines are as low as possible, without changing the separation relations between the basins. If two pixels from different basins were k-separated in the input image, they should still be k-separated in the topological watershed of the image.

The generic algorithm for computing a topological watershed of an image F proceeds by iteratively lowering some point satisfying a condition of destructibility until there is no more such points. More precisely, a point x is said to be W-destructible for F (where W stands for Watershed) if its altitude can be lowered by one without changing the number of connected components of the level set $\overline{F}[k] = \{p \in E; F(p) < k\}$, with $k = F(x)$. A map G is called a W-thinning of F if it may be obtained from F by iteratively selecting a W-destructible point and lowering it by one. A topological watershed of F is a W-thinning of F which contains no W-destructible point.

Both the watershed and a topological watershed of the image from Fig. 1(a) are shown in Fig. 2. Fig. 2(c) shows that the separated pixel pairs from Figure 1(b) are still respectively 5-separated and 3-separated. The third pair is not shown as it was not separated.



(a) (b) (c)

Fig. 2. Watershed and topological watershed: (a) watershed of Fig. 1(a); (b) topological watershed of Fig. 1(a); (c) lowest paths between the separated pairs of pixels from Fig. 1(b)

3 Parallel Implementation

To obtain a fast algorithm, we need to lower a W-destructible point not by 1 as in the generic algorithm, but as much as possible. Following the theoretical results in [4], the lowest value to which a point can be lowered can be computed from the Min-tree $\mathcal{C}(\overline{F})$. This Min-tree, or component tree, can be obtained using the parallel algorithm from [11]. Furthermore, the Min-tree allows one to easily check if two points are separated. More precisely, the separation is related to the altitude of the Lowest Common Ancestor (LCA) between two nodes of $\mathcal{C}(\overline{F})$. For efficiency, we need to perform preprocessing so that the LCA can be found in constant time. This is done using the algorithm in [10]. After this we need to lower all W-destructible pixels in the image to the value of their watershed basin. A pixel is W-destructible if its value can be lowered without linking two local minima into a single basin.

A key element of the algorithm is the function `W-Destructible`, which determines to which grey level a pixel can be lowered in the output. This uses the LCA value of neighbouring components of a the Min-tree. Because the function `W-Destructible` is a local function, we can parallelize the topological watershed algorithm for n threads simply by dividing the image into n tiles and assigning one tile to each thread. An example division for a 2D image is illustrated in Figure 3(a) and a division for a 3D volume is shown in Figure 3(d).

Problems arise when examining border pixels, because their neighbours in other tiles can be changed at any time by their assigned threads, producing incorrect results. We can solve these problems by letting each thread process its tile in different stages, and synchronizing all threads after each stage. Each tile is divided into sub-tiles, and a different sub-tile is processed in each stage. The tiles are divided in such a way that no two adjacent sub-tiles need to be processed at the same time.

Whether or not two sub-tiles are adjacent depends on the connectivity, as shown in Fig. 3. Fig. 3(a) shows an image divided into 12 tiles for 12 threads. With 4-connectivity, each tile should be divided into 2 sub-tiles, as shown in Fig. 3(b). The dark sub-tiles represent the sub-tiles that are processed in the first

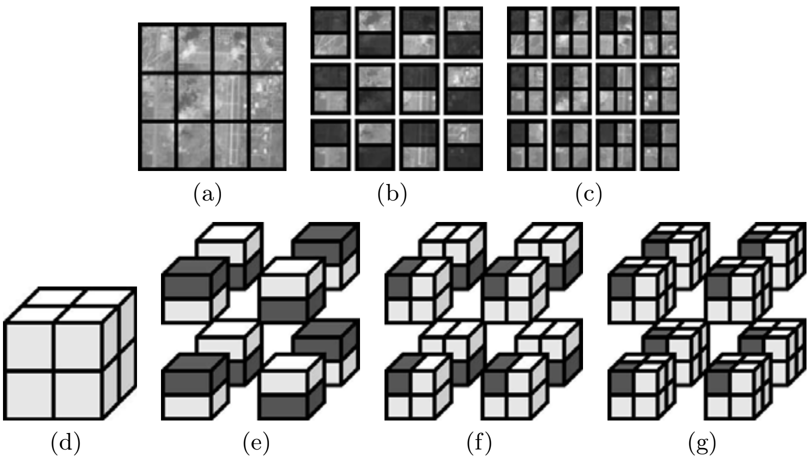


Fig. 3. Tiling: (a) 2D image; (b) tiling for 4-connectivity; (c) same for 8-connectivity; (d) 3D volume; (e) tiling for 6-connectivity; (f) same for 18-connectivity and (g) 26-connectivity;

stage. Note that no two dark sub-tiles are neighbours of each other. However, they would be neighbours with 8-connectivity, so 4 sub-tiles are used for 8-connectivity, as shown in Fig. 3(c). Fig. 3(d) shows a 3D volume divided into 8 tiles for 8 threads. For 6-, 18- and 26-connectivity, 2, 4 and 8 sub-tiles are needed, respectively. This is illustrated in Figures Fig. 3(e), (f) and (g). Again, the dark sub-tiles represent the sub-tiles to be processed in the first stage.

A single application of the multi-stage algorithm suggested above is however insufficient. For example, a certain pixel x may need to be lowered to the value of some local minimum y to obtain a topological watershed of the input image. However, if pixels x and y are part of different sub-tiles, it is possible that pixel x will not get the value of y the first time its sub-tile is processed. Multiple iterations may be needed to obtain the desired result, as in the case of regular watersheds [9].

The sequential topological watershed algorithm [4] processes all pixels in an order that is determined by a priority queue, built as preprocessing stage. In the parallel implementation, this is done when the pixels are processed for the first time, but visiting each pixel in every later iteration is not necessary. Instead, we keep track of the pixels that have been changed, and only add those pixels to the priority queue that are adjacent to pixels in other sub-tiles that changed recently. For this purpose we will use the binary map `pxChanged`, that will store for each pixel whether or not its value has changed recently. The use of this map is illustrated in Fig. 4.

The parallel algorithm is distributed over two procedures: the procedure `InitializeQueue`, which corresponds to the first part of the sequential algorithm, and the procedure `TopologicalWatershedTile`. , which corresponds to the second part. Both procedures are given below.

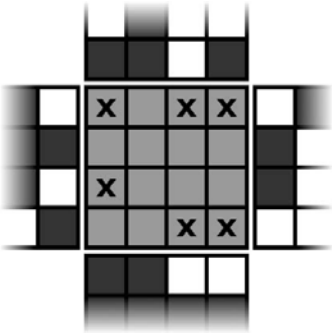


Fig. 4. An example of a `pxChanged` map in an iteration after the first. The white pixels in the four adjacent sub-tiles are pixels that are marked as 'changed' in the `pxChanged` map. The pixels in the current sub tile that are adjacent to such a changed pixel, marked with x in this figure, are added to the priority queue of the sequential algorithm.

As its input, the `InitializeQueue` procedure needs the image F, the min-tree $\mathcal{C}(\overline{F})$ and the corresponding component map Ψ that associates to each point the node it belongs to in $\mathcal{C}(\overline{F})$. Additionally, it also needs the sub-tile T in which it should operate, and the current state of the `pxChanged` map. If the pixel turns out to be W-destructible, it is added to the priority queue with its priority set to the level to which the pixel may be lowered. Also, this new level is stored in the map K and a pointer to the component to which the pixel may be added is stored in the map H. The pixel itself is not lowered yet. Thus, the output consists of the priority queue L, the maps K and H, and the updated `pxChanged` map. The queue L and the maps K and H are all local, but the map `pxChanged` is global, and may be read and modified by other threads while this procedure is being executed. However, each thread will only write in the part of the `pxChanged` map that corresponds to its current sub-tile, and will only read in adjacent sub-tiles that are processed in a different stage, so no conflicts emerge.

The algorithm starts by initializing the priority queue L in line `01`. It then proceeds by setting the `pxChanged` map to `false` for every pixel in sub-tile T. If the procedure is run during the first iteration, then lines `04` to `08` are executed. In the first iteration, L is initialized exactly like in the original sequential algorithm from [4], apart from the fact that only the pixels within T are processed instead of all pixels in F.

If the procedure is called after the first iteration, lines `09` to `18` are executed. In these lines the algorithm checks all border pixels for changed neighbours, and tests the pixels for W-destructibility if any changed neighbours are found. If a pixel turns out to be W-destructible, it is added to the priority queue, and the maps K and H are updated as before.

The second procedure, `TopologicalWatershedTile`, needs the same input as the procedure `InitializeQueue`. The output consists of the updated image F, the updated `pxChanged` map, and the binary variable `anyChanges`. This variable

Algorithm 1. Queue Initialization

```
Procedure InitializeQueue (Input $F, \mathcal{C}(\overline{F}), \Psi$, $T$, $pxChanged$;
Output $L$, $K$, $H$, $pxChanged$)
01. For $k$ From $k_{\min}$ To $k_{\max} - 1$ Do $L_k \leftarrow \emptyset$
02. For All pixels $p \in T$ Do $pxChanged[p] \leftarrow$ false
03. If first iteration Then
04.     For All pixels $p \in T$ Do
05.         $c \leftarrow$ W-Destructible($F, p, \mathcal{C}(\overline{F}), \Psi$)
06.         If $c \neq \emptyset$ Then
07.             $i \leftarrow$ level of $c$; $L_i \leftarrow L_i \cup \{p\}$
08.             $K(p) \leftarrow i; H(p) \leftarrow$ pointer to $c$
09. Else
10.     For All border pixels $p$ of $T$ Do
11.         addP $\leftarrow$ false
12.         For All neighbours $q$ of $p$ Do
13.             If $pxChanged[q]$ = true Then addP $\leftarrow$ true
14.         If addP = true Then
15.             $c \leftarrow$ W-Destructible($F, p, \mathcal{C}(\overline{F}), \Psi$)
16.             If $c \neq \emptyset$ Then
17.                 $i \leftarrow$ level of $c$; $L_i \leftarrow L_i \cup \{p\}$
18.                 $K(p) \leftarrow i; H(p) \leftarrow$ pointer to $c$
```

`anyChanges` is used to quickly determine if any changes have occurred in the sub-tile during the execution of this procedure.

The procedure `TopologicalWatershedTile` starts by initializing the value of `anyChanges`. The function call on line `02` produces an initialized priority queue L, as well as initialized maps K and H. The rest of the function is mostly the same as the second part of the sequential algorithm. Line `08` is added, where the `pxChanged` map and the `anyChanges` variable are updated. Also, a new restriction is added to line `09`, saying that only neighbours of p that lie within the sub-tile T should be added to the priority queue.

With the procedure `TopologicalWatershedTile` implemented, we can now define the main parallel algorithm: the procedure `ParallelTW`. As its input, it needs the pixel mapping F, the Min tree $\mathcal{C}(\overline{F})$ and the corresponding component map Ψ. Because each thread will run this procedure independently, the global map `pxChanged` needs to be provided to each thread as well. However, no initial values need to be stored in it. Additionally, each thread is provided its identifier `id`. The first thread gets `id` value `0`, the second gets value `1` and so on. The output of the procedure is the updated map F, that will contain the topological watershed of the input image.

The algorithm keeps looping until a topological watershed of the input image is found. In each iteration, the loop starting on line `02` is executed once by each thread. Line `04` starts a loop that visits all stages. The number of stages is equal to the number of sub-tiles assigned to each thread, as shown in Fig. 3. Line `05` then determines the location and dimensions of the sub-tile to be processed by the current thread in the current stage. Some examples of the sub-tiles that

Algorithm 2. The algorithm for a single tile

```
Procedure TopologicalWatershedTile (Input $F, \mathcal{C}(\overline{F}), \Psi$, $T$, $pxChanged$;
Output $F$, $pxChanged$, $anyChanges$)
01. $anyChanges \leftarrow$ false
02. InitializeQueue($F, \mathcal{C}(\overline{F}), \Psi$, $T$, $pxChanged$)
03. For $k$ From $k_{\min}$ To $k_{\max} - 1$ Do
04.     While $\exists p \in L_k$ Do
05.         $L_k = L_k \backslash \{p\}$
06.         If $K(p) = k$ Then
07.             $F(p) \leftarrow k; \Psi(p) \leftarrow H(p)$
08.             $pxChanged[p] \leftarrow$ true; $anyChanges \leftarrow$ true
09.             For All neighbours $q$ of $p$ within $T$, with $k < F(q)$ Do
10.                 $c \leftarrow$ W-Destructible($F, q, \mathcal{C}(\overline{F}), \Psi$)
11.                 If $c = \emptyset$ Then $K(q) \leftarrow \infty$
12.                 Else
13.                     $i \leftarrow$ level of $c$
14.                     If $K(q) \neq i$ Then
15.                         $L_i \leftarrow L_i \cup \{q\}; K(q) \leftarrow i$
16.                         $H(q) \leftarrow$ pointer to $c$
```

should be processed by each thread in the first stage are displayed in Fig. 3. In the other stages the sub-tiles that are processed should have a similar pattern, always assuring that no two adjacent sub-tiles are processed at the same time.

The topological watershed of the tile is then computed on line 06. If the algorithm returns that there have been some changes, then this is stored for the current thread in the (global) `anyChangesThr` array. After this, a standard barrier function is called, that just waits until all threads have reached this barrier and then lets all threads continue. This is done to ensure that no thread will start with the next stage until all threads are done with the current one.

When all threads have finished processing all their sub-tiles, the first thread will check if there have been any changes in any of the threads. If there have not been any changes at all, a topological watershed has been found and all threads will terminate. Otherwise, each thread will go to the next iteration by starting again with the main loop from line 02.

3.1 Min-Tree Compression

Wilkinson et al. [11] described how to parallelize the computation of a min-tree. Because the algorithm from [11] also deals with features of the min-tree that we won't use, its implementation is simplified somewhat. Basically, the sequential algorithm is parallelized by letting multiple threads each compute the min-tree of a different part of the input image, and merging the min-trees of the parts afterwards. This algorithm uses a representation of the min-tree which is as large as the image or volume itself, i.e. each pixel or voxel is a node, containing a pointer to its parent. Only those nodes which have a parent with grey level greater than its own are relevant to either filtering or the watershed computation.

Algorithm 3. The parallel topological watershed algorithm

```
Procedure ParallelTW (Input $F, \mathcal{C}(\overline{F}), \Psi$, pxChanged, id; Output $F$)
done ← false
While not done Do
    anyChangesThr[id] ← false
    For All stages s Do
        $T$ ← current sub-tile, based on id and s
        TopologicalWatershedTile($F, \mathcal{C}(\overline{F}), \Psi, T$, pxChanged)
        If anyChanges = true Then anyChangesThr[id] ← true
        Barrier()
    If $id = 0$ Then
        anyChangesAtAll ← false
        For All threads t Do
            If anyChangesThr[t] = true Then
                anyChangesAtAll ← true
    Barrier()
    If anyChangesAtAll = false Then done ← true
```

These nodes are called *level roots*. When the min-tree is built in parallel, not all pixels of a given min-tree node directly point to the level root. This means finding a level root is costly. Because the LCA algorithm inspects the level roots often, we need to compress the tree in the sense that all parent pointers always point to a level root, yielding what is referred to as a *canonical representation* of the tree [6]. This reduces the computation time of the LCA algorithm. The algorithm is shown in Alg. 4.

The procedure `CompressTree` needs a map F, a component tree $\mathcal{C}(\overline{F})$ and a corresponding component map Ψ as its input, as well as the identifier `id` of the thread that executes it. The output of the procedure consists of the compressed component tree and map. Function `LevelRoot` is used by `CompressTree` to obtain the level roots, without ever writing in memory sections not assigned to the current thread.

Algorithm 4. The Min-tree compression algorithm.

```
Function LevelRoot (Input $c$, $F$, $\mathcal{C}(\overline{F})$)
If $c = \text{root}(\mathcal{C}(\overline{F})) \vee F(c) \neq F(\text{parent}(c))$ Then
    Return $c$
Else
    Return LevelRoot(parent($c$), $\mathcal{C}(\overline{F})$, $F$)
```

```
Procedure CompressTree (Input $F$, $\mathcal{C}(\overline{F})$, $\Psi$, id; Output $\mathcal{C}(\overline{F})$, $\Psi$)
For All components $c \in$ segment id of $\mathcal{C}(\overline{F})$ Do
    parent($c$) ← LevelRoot(parent($c$), $F$, $\mathcal{C}(\overline{F})$)
For All pixels $p \in$ segment id of $F$ Do
    $\Psi(p)$ ← LevelRoot($\Psi(p)$, $F$, $\mathcal{C}(\overline{F})$)
```

4 Results

The implementation was tested on the following four different input images: (i) a 4000 × 4000 satellite image of an airfield, (ii) a 4000 × 4000 image in which each pixel has a random grey value (iii) a 1000 × 1000 image with a spiral shaped plateau, and (iv) an angiogram with dimensions 256 × 256 × 128. The spiral image was chosen as a (near) worst-case scenario, in the same way as in [9]. The random noise image was another extreme case. The images are displayed in Fig. 5.

All data sets were tested on a 4-socket, 6-core per sockets, AMD Opteron-based machine with 128 GB of memory, using 1 to 24 threads, and various connectivities (4 and 8 in 2D, 6, 18 and 26 in 3D). The memory is divided into banks assigned equally to each processor sockets, but is accessible as shared memory amongst all cores. Accessing memory of another processor socket does incur a slight speed penalty. In all cases, each thread was initially assigned an equal slice of the image or volume during the min-tree construction phase, as in [11], and an equal tile as described in section in the remainder of the algorithm as detailed above.

The previously existing sequential algorithm by Couprie et al. [4] and the newly implemented parallel algorithm were run on the same machine with the same input (the satellite image, with 4-connectivity using 1 thread) to compare the two versions on a single core.

On the satellite image, we obtained a wall-clock time of 86.36 s at 4 connectivity, for a single thread. This dropped to 17.14 s at 8 threads, 9.71 s at 16 threads, and 7.17 s at 24 threads. At 8 connectivity, all times rose, and the timings were 135.17 s at 1 thread, decreasing in a similar fashion to 12.04 s at 24 threads. The noise image (at 4 connectivity) showed a very similar pattern, decreasing from 112.83 s on 1 thread to 11.95 s on 24 threads. The influence of connectivity was most profound in the 3D case. At 6 connectivity, timings run from 73.18 s at 1 thread to 5.7 s at 24 threads. At 18 connectivity these figures rise to 214.71 s and 16.62 s respectively, increasing further to 359.22 s and 33.37 s at 26 connectivity.

As expected, the (smaller) spiral image behaved very differently, with timings of 4.29 s on 1 thread, 1.71 s on 8, 1.58 s on 16, and increasing again at 24 threads

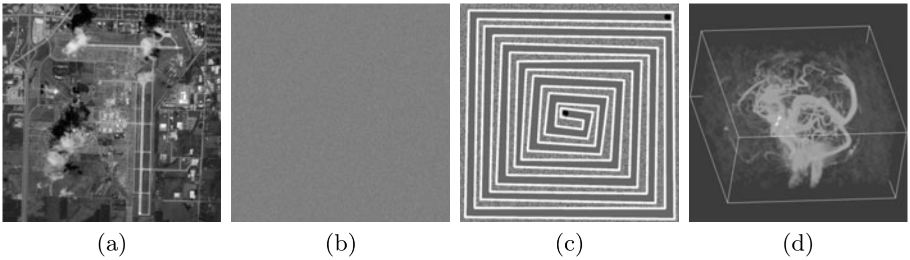


Fig. 5. The four test input images. (a) shows a satellite image, (b) an image with random pixel values, (c) an image with a spiral-shaped plateau and (d) an angiogram, which is a 3D volume.

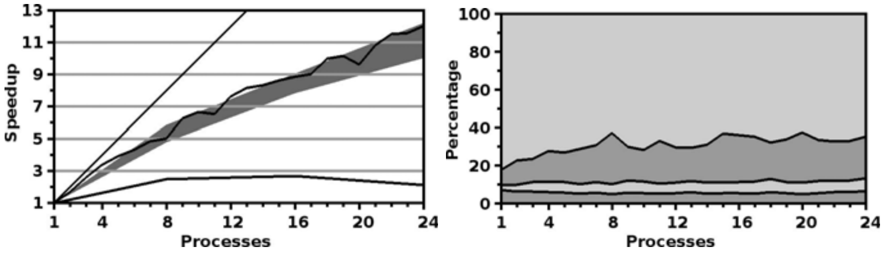
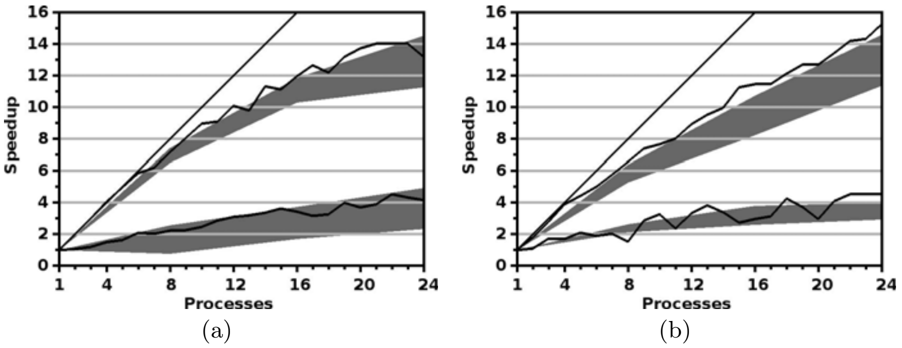


Fig. 6. The overall behaviour of the algorithm: (a) The total speedups in the performed tests, relative to the wall-clock time for a single thread with the same input; (b) the contributions of the four stages to the total wall-clock time of the algorithm, when computing the topological watershed for the satellite image with 4-connectivity.



(a)

(b)

Fig. 7. The speedups of the four stages of the algorithm: (a) Top: min-tree construction, bottom: tree compression; (b) Top: topological watershed computation, bottom: LCA preprocessing

to 1.99 s. This is due to the increase in the number of iterations of the topological watershed algorithm.

The overall behaviour is shown in Fig. 6. In Fig. 6(a) the diagonal line shows the ideal speedup, the line at the bottom shows the speedup for the spiral image. The line through the grey area shows the speedup for the satellite image with 4-connectivity, which is computed for each number of threads individually. The grey area represents the results of the all tests, excluding for the spiral image. The top and bottom of the grey area are defined as the average speedup of their stage plus and minus the standard deviation, respectively.

In Fig. 6(b) we show the contributions of the four stages to the total wall-clock time of the algorithm. The bottom layer represents the time consumed by the construction of the min-tree, the second layer from the bottom represents the time taken to compress the min-tree, the third layer shows the time it takes to perform the preprocessing for the LCA, and finally the remaining layer on top represents the time consumed by the final stage which produces the topological watershed.

The individual speedups are shown in Figure 7. The speedup of each stage is represented by a black line and a grey area, where the line shows the speedup of that stage when computing the satellite image with 4-connectivity. The corresponding grey area represents the results of the remaining tests as in Fig. 6, again excluding the spiral image. The diagonal lines show the ideal speedup, where the speedup is equal to the number of threads.

The construction of the component tree and the final stage that computes the topological watershed parallelize quite well, while the tree compression and the LCA preprocessing parallelize rather poorly. Fortunately, the tree compression takes up only a small percentage of the total wall-clock time, even when 24 threads are used. However, this percentage will probably increase when more threads are used. The LCA has a larger impact on the total wall-clock time of the algorithm, and will cause the speedup of the total algorithm to decrease more severely when using a larger number of threads.

Note however, that all speedups reported are relative to the wall-clock times of the parallel algorithm using one thread. In practice, the previously existing sequential implementation performs about 1.5 times faster than parallel algorithm when only one thread is used. On two threads, the current algorithm has the edge, albeit by a small margin. Further optimizations in the parallel algorithm, for example by using a better tree compression method, may reduce this difference.

5 Conclusions

This paper described a way to parallelize the computation of the topological watershed. An implementation that was created according to this description showed that a reasonably good speedup could be achieved while using up to 24 threads, and the trend in the results suggests that even better speedups may be achieved when more than 24 threads are used.

The parallel implementation may be improved further by creating a more compact min-tree in the min-tree construction or compressing it more in the tree compression stage. For example, each node in the min-tree that has only one child can be merged with that child, setting all pixels belonging to its component to the grey level of its child, and setting the parent of its child to be the parent of the node itself. This simple change could significantly reduce the LCA preprocessing time, and may also improve the wall-clock time of the last stage where the topological watershed is computed.

Better results may also be obtained by implementing the LCA preprocessing more efficiently, especially the list ranking part (see [10]). A simple list ranking algorithm was used in our implementation, while more sophisticated and better performing algorithms already exist. A list ranking algorithm that parallelizes better and has higher speedups, may significantly reduce the total wall-clock time of the algorithm, especially when using larger numbers of threads.

Furthermore, the image is now divided into tiles which are each processed by their own thread. This division is only based on the image dimensions and the number of threads, not on the contents of the image. Taking into account the contents of the image, maybe in combination with the min-tree and component map, while dividing the image among the threads may reduce the communications needed between the different threads, which could lead to a faster algorithm. We can also try other ways of tiling the image, following [5].

In short, there is still room for improvement in the parallel algorithm proposed in this paper, but in its current form it can already be used to greatly speed up the computation of the topological watershed, compared to the previously existing sequential algorithm.

References

1. Bertrand, G.: On topological watersheds. Journal of Mathematical Imaging and Vision 22(2), 217–230 (2005)
2. Beucher, S., Lantuéjoul, C.: Use of watersheds in contour detection. In: International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation, pp. 17–21 (1979)
3. Couprie, M., Bertrand, G.: Topological grayscale watershed transformation. In: SPIE Vision Geometry V Proceedings, vol. 3168, pp. 136–146. Citeseer (1997)
4. Couprie, M., Najman, L., Bertrand, G.: Quasi-linear algorithms for the topological watershed. J. Math. Imag. Vis. 22, 231–249 (2005)
5. Matas, P., Dokládalová, E., Akil, M., Grandpierre, T., Najman, L., Poupa, M., Georgiev, V.: Parallel Algorithm for Concurrent Computation of Connected Component Tree. In: Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2008. LNCS, vol. 5259, pp. 230–241. Springer, Heidelberg (2008)
6. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Trans. Image Proc. 15, 3531–3539 (2006)
7. Najman, L.: On the equivalence between hierarchical segmentations and ultrametric watersheds. J. Math. Imag. Vis. (2010) (to appear), http://www.laurentnajman.org
8. Najman, L., Couprie, M., Bertrand, G.: Watersheds, mosaics and the emergence paradigm. Discrete Applied Mathematics 147(2-3), 301–324 (2005)
9. Roerdink, J.B.T.M., Meijster, A.: The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. Fundamenta Informaticae 41, 187–228 (2001)
10. Schieber, B., Vishkin, U.: On finding lowest common ancestors: simplification and parallelization. SIAM Journal on Computing 17(6), 1253–1262 (1988)
11. Wilkinson, M.H.F., Gao, H., Hesselink, W.H., Jonker, J.E., Meijster, A.: Concurrent computation of attribute filters using shared memory parallel machines. IEEE Trans. Pattern Anal. Mach. Intell. 30(10), 1800–1813 (2008)

Advances on Watershed Processing on GPU Architecture

André Körbes[1], Giovani Bernardes Vitor[2],
Roberto de Alencar Lotufo[1], and Janito Vaqueiro Ferreira[2]

[1] University of Campinas, School of Electrical and Computer Engineering
(Department of Computer Engineering and Industrial Automation),
Campinas - SP, Brazil,
{korbes,lotufo}@dca.fee.unicamp.br
[2] University of Campinas, School of Mechanical Engineering, Campinas - SP, Brazil
{giovani,janito}@fem.unicamp.br

Abstract. This paper presents an overview on the advances of watershed processing algorithms executed on GPU architecture. The programming model, memory hierarchy and restrictions are discussed, and its influence on image processing algorithms detailed. The recently proposed algorithms of watershed transform for GPU computation are examined and briefly described. Its implementations are analyzed in depth and evaluations are made to compare them both on the GPU, against a CPU version and on two different GPU cards.

1 Introduction

The watershed transform is a well known image segmentation tool, used to compute connected components, generally out of the gradient of the input image. Several algorithms exist to process the transform sequentially [4], [8], and since the development of the first fast transforms, effort is put on research of parallel algorithms and architectures. This has led to the recent development of techniques used on clusters [1] and FPGA [9].

The rise of GPU cards programmable for generic purposes, with low cost and high computational power, and with a trend of improvement of the hardware without severe changes to the architecture and programming model, has established a new platform of interest. These features sets the direction of advances on the watershed algorithms for improving performance, using the concepts and definitions established on literature, exploring the architecture for the procedures that best fit, and showing some new and revisited approaches [3], [13], [12].

This paper works on studying this architecture and the advances made on the watershed processing and the algorithms that use it. The usage of such models influences how the problems must be treated, algorithms designed and programs implemented. The watershed transform is implemented on two forms: one with four steps, with different levels of division of tasks, ranging from the neighborhood to global processing; and one that uses only a neighborhood operation,

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 260–271, 2011.

switching values until the optimal solution is found and stabilizes. These different levels of decomposition are analyzed, and the implementation strategies described.

This paper is organized as follows: Sec. 2 discusses the influence of the GPU architecture on the development of image processing algorithms; Sec. 3 reviews the literature of parallel watershed algorithms and discusses the two algorithms evaluated in this work; Sec. 4 discusses in details the implementation of both algorithms; Sec. 5 discusses the measurements made, and exposes the performance obtained on two GPU cards and a CPU; and lastly, Sec. 6 presents the conclusions drawn from this work.

2 Architecture Influence

Given many threads executing, each one having a small computing power, with some shared resources and high bandwidth for communication, the GPU architecture is well suited for parallel algorithms of fine granularity, where each thread will process one or a few pixels of the output image. Regarding the communication between threads, as they share a common memory, this area is used when any kind of information needs to be transferred between them. Also, whenever using GPU memory to communicate with the host CPU, there is a considerable overhead on the copy operation that may degrade the speedup obtained.

As many image processing algorithms, the watershed transform processes a neighborhood around every pixel. Also, as the blocks of threads divide the image on sub-images that are loaded on shared memory, these must have a border with the pixels from the adjacent blocks. This concept is illustrated on [7] as the apron for each block. The apron is constituted such that blocks are processed with overlapped data, with each block responsible for processing an area of the image, but loading data from adjacent blocks.

The modern GPU architecture evolved to a point where there are several forms of running programs on it, with the most noticeable frameworks and languages being CUDA and OpenCL, which are very similar to C and C++. Both models provide access to the same memory model with similar instructions. The programs developed for this work used CUDA, and the source codes are available on the Internet at http://www.adessowiki.org

3 Parallel Watershed Algorithms

The watershed transform is a very data intensive task, that even with quasi-linear algorithms is time-consuming. Since its introduction by Vincent and Soille [10] effort is put on researching faster algorithms as well as parallel approaches. An extensive survey on parallel approaches is given by Roerdink and Meijster [8]. Since then, faster algorithms have been created, as well as new parallel approaches [9], [1]. On the past few years, other parallel watershed algorithms have been introduced, specifically for GPU architectures [3], [13], [12], using different strategies, designed for the restrictions of the platform.

This paper analyzes two algorithms: one inspired by the drop of water paradigm and depth-search approaches, based on [12], named **DW**; and one based on cellular automata to process a shortest-path forest with sum cost function [3], named **CA**. Both algorithms are tuned to better suit the architecture of modern GPUs.

The **DW** algorithm uses different steps of processing, which allows faster processing of operations with different levels of data independence. These steps are briefly: (i) identification of the neighbor with the lowest gray level for each pixel (this corresponds to finding the downstream for each pixel, see [6], [8]); (ii) propagation of the downstream of the plateau border pixels to inner pixels according to the geodesic distance; (iii) labeling of regional minima by union-find; and (iv) labeling of non-minima pixels by the path created to every minima. These steps are organized and designed to maximize performance on a GPU architecture. Step (i) processes each pixel independently, scanning the neighborhood for the lowest gray level; step (ii) is the step less suited for parallelism, caused by the need to uniformly calculate the geodesic distances, creating synchronization barriers on the threads; step (iii) consists of labeling the regional minima pixels using the union-find strategy for merging connected components; step (iv) consists on compressing paths from each pixel to the regional minima associated with the path.

The **CA** algorithm is based on the algorithm of Ford-Bellmann to calculate a shortest-path forest, using a single relaxing procedure, performed until convergence of the solution. Therefore, the proposed implementation consists of a single step, executed until stabilization of the solution. Nevertheless, for the **CA** algorithm produce a correct watersheds transform, its cost function must be adapted to consider the weight of each edge as the topographic distance [6]. Also, as this function does not manage plateaus, either the image must be preprocessed by lower completion, or the cost function must be further modified to consider the lexicographic cost as a secondary component [5]. For greater adherence with the original proposal, the lower completion is used prior to watershed execution.

4 Implementation Details

This section discuss the implementation of watershed algorithms on the SPMD and SIMD models, considering the access to the memory layers of the GPU, that constitutes the problem of communication between blocks. It is also presented the methodology used for development, highlighting the border treatment, the replication of data, and the data flow between memories.

The GPU memory model is hierarchically divided, and programs must consider these different levels and its characteristics to take advantage of its features, ultimately to reach the best performance. Using the CUDA framework, the code that is executed on the GPU does not have access to the CPU RAM memory, being an area of access exclusive of the host code. Therefore, for any processing on the GPU, the data must be transferred between host (CPU) memory and device (GPU) memory. The GPU global memory is the area writeable by the CPU, and it also provides special access modes of texture and constant, which are cached and only readable by the device code. There are three other memory areas available to the device code: the registers, the local and the shared memory. These memories have very limited size, but are located close to each stream multiprocessor of the GPU, and provide high speed of access. Registers and local memory are bound to the scope of each thread, whereas the shared memory, with scope on the block, is used for cooperation and communication between threads, with also a high speed access.

Based on the communication relations between different types of memory, the host and device code that composes the watershed processing steps are modeled to use as most as possible the shared memory area. Initially the image is loaded by the host on a texture memory, and the neighborhood relation data on the constant memory. As the blocks process parts of the image, it is loaded to the shared memory. From this, the results are computed on registers, and finally written on the global memory. This flow is used on all kernels of the algorithms developed, except those that do not use a neighborhood relation, where the global memory is used directly.

With the decomposition of the image on sub-images arises the problem of management of the borders of each block, as each pixel demands data from the neighbors that may be contained on another block, to complete its computation. The strategy used is to load an overlapping area, called the apron, as described on [11], [7], according to the neighborhood relation used. The borders values are either the values of pixels of the adjacent block, or in the case of image borders, a constant and predefined mask value is used.

The overlapping scheme, with borders loaded for every block by the device code, is presented on Fig. 1. This scheme is divided on two phases: the first one loads the data to the shared memory, while the second processes the shared memory and stores the results back on global memory. The loading is performed on blocks of $BLOCK\_TILE$ width and height, which corresponds to the size of blocks of threads used on the CUDA device code invoking. The processing is performed on fewer threads, on a block of $REAL\_TILE$ width and height, which corresponds to the dimensions of the block of data that is actually processed, without overlapping, ignoring border pixels.

4.1 Algorithm DW

The **DW** algorithm is based on the four steps described on Sec. 3 and extensively detailed on [12]. To implement this algorithm, six programs of device code (kernels) were created:

- *donwstreamCalc*: calculates the downstream for each pixel and then uses the data already loaded on memory to propagate it to plateaus internal pixels, until stability inside the block. This kernel is executed once and is associated to the first and second steps of the algorithm.
- *plateauPropagate*: continues the propagation of the downstream to plateaus internal pixels inside the block, respecting the geodesic distance to the borders. As the distance propagation may require block communication, this kernel is invoked until stabilization and is associated with the second step.

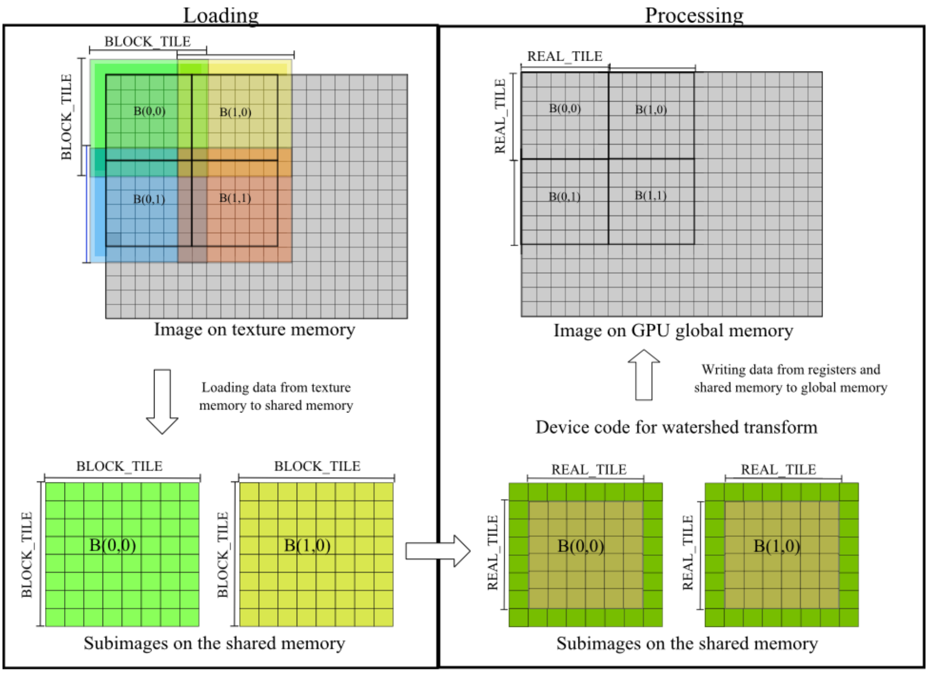

Loading
Processing
BLOCK_TILE
BLOCK_TILE
B(0,0)
B(1,0)
B(0,1)
B(1,1)
Image on texture memory
Loading data from texture memory to shared memory
BLOCK_TILE
BLOCK_TILE
BLOCK_TILE
BLOCK_TILE
B(0,0)
B(1,0)
Subimages on the shared memory
REAL_TILE
REAL_TILE
B(0,0)
B(1,0)
B(0,1)
B(1,1)
Image on GPU global memory
Writing data from registers and shared memory to global memory
Device code for watershed transform
REAL_TILE
REAL_TILE
REAL_TILE
REAL_TILE
B(0,0)
B(1,0)
Subimages on the shared memory

Fig. 1. Overlapping of data loaded on shared memory to process border pixels correctly. It is used extra threads inside each block to load data, which are discarded and do not perform the computation.

- *mergeRegions*: creates connected components with the minima pixels, using the index as the reference for the group. The regional minima may spread across different blocks, so this kernel is invoked until stability, paired with the pathCompression kernel. This kernel is associated with the third step of the algorithm.
- *pathCompression*: updates the reference of each pixel directly to the connected component root. This avoids unnecessary memory accesses when labeling the image and merges connected components that spread across blocks. This kernel is associated with the third and fourth steps of the algorithm. On the third step this is invoked until stabilization of the solution, and only once for the fourth step.
- *labelPixel*: updates the label of each pixel, according to the reference of its group. For regional minima processing, this connects minima that spread on more than a single block, unifying the labels. This kernel is associated with the third step of the algorithm.
- *indexAdjust*: updates the references matrix to allow the pathCompression kernel to work on all pixels, rather than only on the regional minima, as a preparation for the fourth step.

The minima and paths labeling problem is addressed in many ways on both sequential and parallel algorithms, being a usual bottleneck, where common strategies have not obtained success on the GPU [12]. For that matter, these steps on the current approach are performed using the labeling algorithm of [2] , based on a reference list for path compressing and representative propagation. This algorithm is implemented through the kernels *mergeRegions*, *pathCompression* and *labelPixel* presented before.

To better understand the bottlenecks of this algorithm, each kernel and API call is timed, and a profile of the amount of time spent on each step is produced, shown on Fig. 2. The first kernel executed, downstreamCalc, takes around 60% of the total time of execution. This is expected, as this step calculates both the downstream and solves small plateaus that are contained inside a block. The next kernel invokes, for plateau propagation, take around 16% of the total time, with 8 calls, each with a smaller computation time, until stabilization. The regional minima and catchment basin labeling are solved with a small percentage of the total time, indicating an algorithm highly adapted for the task in parallel. The first kernel may look as a severe bottleneck, compared to the second. However, it was observed that this kernel solves many small plateaus, reducing the amount of work to be done on the plateau propagation kernel.

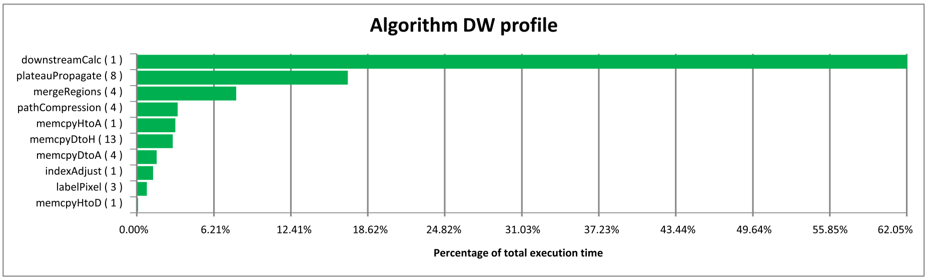


Fig. 2. Percentage of the total execution time taken by each kernel and API call of the DW algorithm

Clearly the division of the algorithm on steps described on Sec. 3 does not match exactly the kernels described on this section, as these are designed to minimize the amount of time consumed on memory copies and maximize the processing on each pixel on a single call. However, those steps divide the problem logically, leading to other solutions, such as the sequential algorithm that is used on the comparisons of the next section. This implementation is also based on the drop of water principle and is divided on the aforementioned steps, and is not parallelized on CPU threads, thus running on a single core. The solutions provided by both algorithms are equivalent and valid.

4.2 Algorithm CA

The **CA** is based on the algorithm described on Sec. 3 and detailed on the work of Kauffmann and Piche [3]. This algorithm, inspired on the algorithm of Ford-Bellmann for calculating shortest-path forests with the sum path cost function, iterates performing changes on the solution whenever a path of less cost is found, until stabilization. To apply this concept on the watershed transform, the weight of each edge of the inner graph must be associated with the topographical distance, as described by Meyer [6]. To address the issue of the cost on each edge depending on the minimal value of the neighborhood, the implementation is based on two kernels:

- *initialization*: initializes the cost for each pixel, depending on whether they are a regional minima or not. Also, scans the neighborhood to store the minimal value, necessary to compute the cost on each edge, during the processing stage.
- *iterate*: process each pixel looking for paths of less cost. This kernel process each block until stabilization of the sub-image solution, and also is invoked as many times as necessary for stabilization of the whole image solution.

To use this algorithm, it is necessary to provide markers, either defined by the user or the regional minima. However, as the kernel that propagates the solution is invoked as many times as necessary, the greatest path on the inner graph will bind the execution time. Thus, the performance is proportional to the ratio of the area of catchment basins by the area of regional minima. To implement this algorithm efficiently, the greatest improvement is the preprocessing of the cost, which leads to several less memory accesses.

5 Performance Measurements

This section presents the experiments performed with the watershed algorithms discussed on this paper, divided on three subsections: first, the algorithms are measured and compared one against the other on the GPU; second, algorithm **DW** is compared against a sequential implementation also based on the drop of water principle; third the algorithm **DW** is timed on two different GPU cards, to evaluate the impact of the evolution of hardware.

The hardware used is: an NVIDIA GTX 295 GPU card with 240 cores running CUDA 2.30; an NVIDA GTX 470 GPU card with 448 cores running CUDA 3.20; and a CPU AMD Phenom II X3 CPU of 2.6Ghz clock and 7.5Mb of cache with 4 GB of RAM. The measurements of GPU algorithm execution also consider memory transfer from CPU to GPU. For the **CA** algorithm, the computation of regional minima is discarded. This computation is not measured as it is out of scope of this work and would implicate on an additional processing that may not be useful if it is supplied markers obtained from other methods, such as user input. The algorithms were run on the images lena, cameraman, peppers and baboon. These images have different profiles, with varying number of regional minima and extension of plateaus. The measurements were averaged, and the standard deviation presented indicates the variation on each image size. The images were resized to 64x64, 128x128, 256x256, 512x512, 1024x1024 and 2048x2048.

5.1 GPU Comparison

This experiment intends to compare the algorithms **DW** and **CA** executing on the GTX 295 card. Tab. 1 shows the average times obtained for each image size, and the standard deviation (STD) for both algorithms. Clearly, the algorithm **DW** produces results much faster and with smaller variation of time. The large standard deviation observed for **CA** is a direct consequence of its single-step propagation that will take as long to execute as the longer path on the image.

Table 1. Comparison of times of algorithm DW and CA running on GTX295

Image size	Alg. DW (ms)	STD	Alg. CA (ms)	STD
64x64	1.065	0.064	1.099	0.094
128x128	1.224	0.126	1.871	0.320
256x256	2.431	0.624	4.872	0.946
512x512	5.801	1.035	19.290	5.224
1024x1024	19.918	3.866	109.532	33.655
2048x2048	88.262	23.322	657.115	253.307

The data of Tab. 1 is also shown as a chart, on Fig. 3, where the difference of times measured is better visualized. This chart presents the measured times in milliseconds (ms) by the total number of pixels on the image.

The results of this experiment may also be compared with the results of the algorithm of Wagner and Godehardt [13], which was executed on a GTX280,

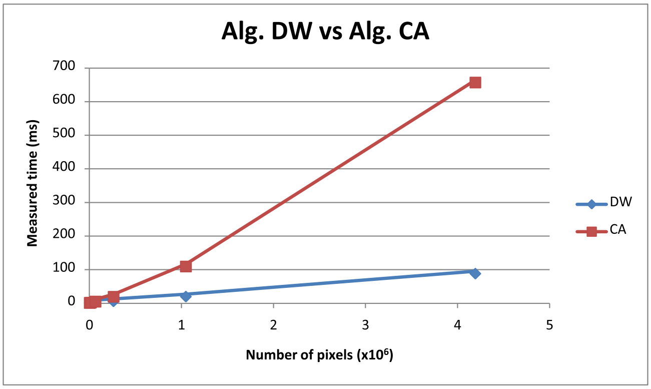


Fig. 3. Chart comparison between algorithms DW and CA running on GTX295

a card very similar to the GTX295. This comparison is possible only for the 1024x1024 image size, which has nearly the same number of pixels of the cube of dimension 100 reported by the authors, with an execution time of 550ms.

5.2 GPU and CPU Comparison

This experiment compares the algorithm **DW** running on the GTX 470 card against the CPU version. The measurements obtained on the executions are shown on Tab. 2. The speedup calculated is the ratio of the CPU by the GPU time. It is seen that for images smaller than 256x256, the speedup is less than two, and given that the absolute times measured are very small, the usage of a GPU algorithm may not prove useful. However, for images larger than that, the speedup is increased up to 6.5, and the absolute times make this acceleration even more useful, which can be seen on the chart of Fig. 4.

Table 2. Comparison of times between GPU GTX470 and CPU implementations

Image size	**GPU (ms)**	**STD**	**CPU (ms)**	**STD**	**Speedup**
64x64	0.646	0.043	0.311	0.003	0.481
128x128	0.776	0.068	1.237	0.059	1.594
256x256	1.213	0.156	4.838	0.337	3.988
512x512	3.365	0.285	18.744	1.414	5.571
1024x1024	10.889	0.873	72.087	3.510	6.620
2048x2048	46.487	8.200	307.067	14.208	6.605

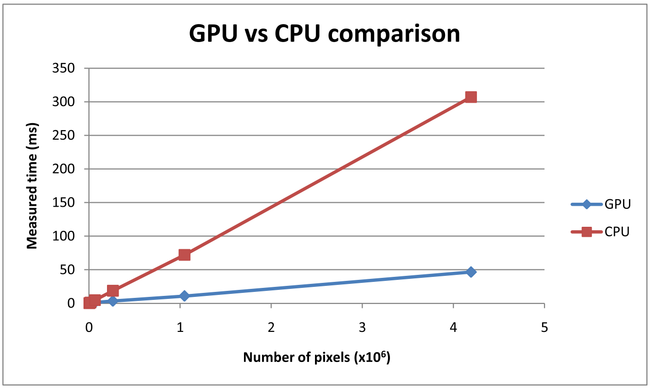


Fig. 4. Chart comparison of times between GPU and CPU implementations

5.3 GPU Evolution Comparison

This experiment intends to show how the evolution of GPU cards impact the execution of the same code. On Tab. 3 it is shown the execution times of the algorithm

Table 3. Comparison of times between two different GPU cards

Image size	GTX 295 (ms)	STD	GTX 470 (ms)	STD	Speedup
64x64	1.065	0.064	0.646	0.043	1.649
128x128	1.224	0.126	0.776	0.068	1.578
256x256	2.431	0.624	1.213	0.156	2.004
512x512	5.801	1.035	3.365	0.285	1.724
1024x1024	19.918	3.866	10.889	0.873	1.829
2048x2048	88.262	23.322	46.487	8.200	1.899

DW on both cards. The speedup is calculated as the ratio of the measurement on the GTX 295 by the timing of GTX 470. The average of the speedup of the images is 1.780. This might be associated with the increase of the number of cores from 240 to 448, which gives a ratio of 1.86. However, it must be noted that there are several other improvements on the architecture of GTX 470, such as enhancing cache capabilities and the operation of atomic functions.

5.4 Algorithm Scalability Comparison

This experiment focuses on how the algorithms **DW** and **CA** perform on the same images, filtered to have less regional minima. To evaluate this, the four images were used on the size of 512x512 and filtered to have different number of regional minima, ranging from 562 to 24848. However, as the images have different profiles, only the image **Lena** is used to show the effect of filtering, as its results were the most affected. The times measured are presented on the chart of Fig. 5. This chart shows the measured times by the number of regional minima. This chart confirms the first experiment, that algorithm **DW** is faster that algorithm **CA**, and shows that the algorithm **DW** is more stable, with

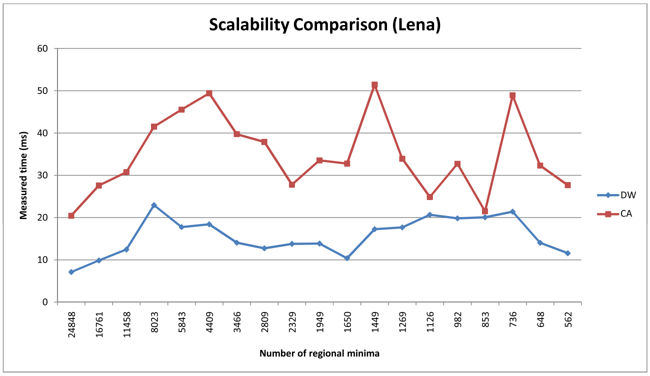


Fig. 5. Chart comparison of measured times by the number of regional minima on image Lena of size 512x512

most times bound between 10 and 20 ms, with a standard deviation of 4.4 ms, while algorithm **CA** is bound between 20 and 50 ms, with a standard deviation of 9.3 ms. The effect of severe variation of running time observed on algorithm **CA** is not observed for every image tested. Nevertheless, such variation was not observed for algorithm **DW** for any of the images.

6 Conclusion

This paper presented an analysis of modern watershed algorithms designed for GPU architectures with some considerations on the implementation and design of such algorithms. The issues of implementing a parallel watershed algorithm were discussed, and two implementations using the CUDA framework were detailed.

On the performance measurements, three scenarios were used, comparing algorithms on the GPU, the GPU against the CPU and GPU cards of different configurations. The comparison on the GPU showed that having steps specialized for each task, and less dependence on stabilization produces faster results. The comparison with the CPU showed that studies of GPU algorithms - i.e. algorithms that use the SPMD model - may lead to reasonable speedups. The comparison of evolution of cards showed that using the number of cores of a board is a reasonable measure of normalization of times, and that the evolution of hardware may further reduce the execution times and restrictions that currently exist.

Also, it has been observed that, because of the levels of speedup achieved, especially when considering the evolution of the GPUs, the cost of implementation of these algorithms is rewarded. In fact, the usage of CUDA and/or OpenCL technologies, with comprehension of the general architecture, enables the development of programs that are not strictly bound to the hardware, and may run on several cards with the benefits of speedup. As a consequence, future works are seen on the investigation of more algorithms focused on the SPMD model, with implementations not dependent on special hardware configuration and also suitable for 3D volumes.

References

1. Galilée, B., Mamalet, F., Renaudin, M., Coulon, P.-Y.: Parallel asynchronous watershed algorithm-architecture. IEEE Transactions on Parallel and Distributed Systems 18(1), 44–56 (2007)
2. Hawick, K.A., Leist, A., Playne, D.P.: Parallel graph component labelling with GPUs and CUDA. Parallel Computing 36(12), 655–678 (2010)
3. Kauffmann, C., Piche, N.: A cellular automaton for ultra-fast watershed transform on GPU. In: 19th International Conference on Pattern Recognition, pp. 1–4. IEEE Computer Society, Tampa (2008)
4. Körbes, A., Lotufo, R.: Analysis of the watershed algorithms based on the breadth-first and depth-first exploring methods. In: SIBGRAPI 2009, pp. 133–140. IEEE Computer Society, Rio de Janeiro (2009)

5. Körbes, A., Lotufo, R.: On watershed transform: Plateau treatment and influence of the different definitions in real applications. In: Proceedings of the 17th International Conference on Systems, Signals and Image Processing, Rio de Janeiro, Brazil, pp. 376–379 (2010)
6. Meyer, F.: Topographic distance and watershed lines. Signal Processing 38(1), 113–125 (1994)
7. Podlozhnyuk, V.: Image Convolution with CUDA. NVIDIA (June 2007), http://developer.download.nvidia.com/compute/cuda/sdk/website/projects/convolutionSeparable/doc/convolutionSeparable.pdf
8. Roerdink, J.B.T.M., Meijster, A.: The watershed transform: definitions, algorithms and parallelization strategies. Fundam. Inf. 41(1-2), 187–228 (2000)
9. Trieu, D.B.K., Maruyama, T.: Real-time image segmentation based on a parallel and pipelined watershed algorithm. Journal of Real-Time Image Processing 2(4), 319–329 (2007)
10. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(6), 583–598 (1991), doi:10.1109/34.87344
11. Vitor, G.: Rastreamento de alvo móvel em mono–visão aplicado no sistema de navegação autônoma utilizando GPU. Master's thesis, Universidade Estadual de Campinas, Campinas, São Paulo, Brazil (March 2010)
12. Vitor, G., Ferreira, J., Körbes, A.: Fast image segmentation by watershed transform on graphical hardware. In: Proceedings of the 30°CILAMCE. Armação dos Búzios, Brazil (November 2009)
13. Wagner, B., Godehardt, M.: Cell reconstruction on stream computing architectures. In: ISMM 2009 Abstract Book, pp. 45–48. University of Groningen (2009)

Incremental Algorithm for Hierarchical Minimum Spanning Forests and Saliency of Watershed Cuts*

Jean Cousty and Laurent Najman

Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, A3SI, ESIEE
{j.cousty,l.najman}@esiee.fr

Abstract. We study hierarchical segmentations that are optimal in the sense of minimal spanning forests of the original image. We introduce a region-merging operation called *uprooting*, and we prove that optimal hierarchical segmentations are equivalent to the ones given by uprooting a watershed-cut based segmentation. Based on those theoretical results, we propose an efficient algorithm to compute such hierarchies, as well as the first saliency map algorithm *compatible* with the morphological filtering framework.

1 Introduction

We study some optimality properties of hierarchical segmentations ([19–21, 5, 23, 16, 8, 24, 2]) in the framework of edge-weighted graphs, where the cost of an edge is given by a dissimilarity between two points of an image. Since the pioneering work of [17, 15] stating an equivalence between hierarchies and minimum spanning trees (MST), numerous hierarchical schemes rely on the construction of such a tree. Its first appearance for classification in pattern recognition dates from the seminal work of Zahn [31]. Its use for image segmentation was introduced by Morris *et al.* [22] in 1986 and popularized in 2004 by Felzenswalb and Huttenlocher [13]. Meyer was the first to explicitly use it in a morphological context [19]. In mathematical morphology, hierarchies of watershed regions have been proposed notably in [5, 23, 20] and reviewed in [21].

In this paper, we formalize, in the framework of edge-weighted graph, a fundamental hierarchical scheme proposed in morphology. This formalism allows us to prove strong properties linking hierarchical segmentations and combinatorial optimality with respect to the original image (Th. 4 and 9). We derive from those properties the first saliency algorithm proved to be correct, establishing in particular its compatibility with morphological filtering (Th. 11). In contrast, all previously proposed algorithms (*e.g.* [5, 23, 18]) are heuristic by nature, and to date, no property on their result has been proved. With those previous algorithms, counter-intuitive results are often obtained [26, 11].

After reminding basic notions in Section 2, hierarchies of minimum spanning forests are presented in Section 3. Then, we formalize, in Section 4, a fundamental

* This work received funding from the Agence Nationale de la Recherche, contract ANR-2010-BLAN-0205-01.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 272–283, 2011.

operation called uprooting that merges a marked region with one of its neighbors linked by the cheapest cost. When applied sequentially on the weighted graph of neighboring regions, the uprootings build a MST of this neighboring graph. Intuitively, one can see that, if one starts from a minimum spanning forest (MSF) rooted in the minima of the image (or, equivalently, from a watershed cut [9]), then one builds a hierarchy of MSFs of the original image itself, the last uprooting step yielding an MST of this original image. More surprisingly, Th. 4 establishes that the two processes are equivalent. Hence, any MST of the original image can be built from an uprooting sequence on a watershed cut. Thus, watershed cuts are the only family of watersheds that allows us to build hierarchical segmentations that are optimal with respect to the original image, in the sense that they preserve the MST of the original image. Based on those results, we give a detailed description of a fast uprooting algorithm that allows MSF hierarchies to be computed. The time-complexity of this algorithm is analyzed. Then, in Section 5, a quasi linear-time algorithm is proposed to compute saliency maps from MSF hierarchies. The correctness of this algorithm is established by Theorem 9. Finally, in Section 6, the compatibility between the MSF hierarchies and a morphological filtering is established by an equivalence result (Theorem 11).

2 Basic Notions

We define a (undirected) *graph* as a pair $X = (V, E)$ where V is a finite set and E is composed of unordered pairs of V, *i.e.*, E is a subset of $\{\{x, y\} \subseteq V \mid x \neq y\}$. If $X = (V, E)$ is a graph, each element of V is called a *vertex or a point of* X, and each element of E is called an *edge of* X.

Let X be a graph, the vertex set and edge set of X are denoted by $V(X)$ and $E(X)$ respectively. The graph X is *nontrivial* whenever $E(X) \neq \emptyset$, and it is nonempty whenever $V(X) \neq \emptyset$.

Let X and Y be two graphs. If $V(Y) \subseteq V(X)$ and $E(Y) \subseteq E(X)$, we say that Y *is a subgraph of* X and we write $Y \sqsubseteq X$. Let $\mathcal{S}$ be a set of graphs. The *supremum (resp. infimum) of* $\mathcal{S}$, denoted by $\sqcup\mathcal{S}$ (resp. $\sqcap\mathcal{S}$), is the graph whose vertex set and edge set are the unions (resp. intersections) of the vertex sets and of the edge sets of the graphs in $\mathcal{S}$: $\sqcup\mathcal{S} = (\cup\{V(X) \mid X \in \mathcal{S}\}, \cup\{E(X) \mid X \in \mathcal{S}\})$ (resp. $\sqcap\mathcal{S} = (\cap\{V(X) \mid X \in \mathcal{S}\}, \cap\{E(X) \mid X \in \mathcal{S}\})$). We also write $X \sqcup Y$ and $X \sqcap Y$ for respectively $\sqcup\{X, Y\}$ and $\sqcap\{X, Y\}$.

Let X be a graph. The graph X *is connected* if it cannot be partitioned into two nonempty graphs, *i.e.*, for any two nonempty graphs Y and Z, if $Y \sqcup Z = X$, then the graph $Y \sqcap Z$ is necessarily nonempty. Let Y be a graph. We say that Y is a *connected component of* X, or simply a *component of* X, if Y is a subgraph of X that is connected and that is maximal for this property, *i.e.*, for any connected graph Z, if $Y \sqsubseteq Z \sqsubseteq X$, then we have necessarily $Z = Y$. The set of all connected components of X is denoted by $\mathcal{CC}(X)$.

Important notation. *Throughout this paper* $G = (V, E)$ *denotes a nontrivial connected graph.*

If S is a subset of V (resp. E), we denote by $\overline{S}$ *the complementary set of S in V (resp. E), i.e.*, $\overline{S} = V \setminus S$ (resp. $\overline{S} = E \setminus S$).

Let $S \subseteq E$, we denote by $\delta^\bullet(S)$ the set of all vertices in V that belong to an edge in S. Remark that $(\delta^\bullet(S), S)$ is a graph (see [12] for morphological properties of $\delta^\bullet$). This graph $(\delta^\bullet(S), S)$ is called the *graph induced by S*.

In the following, the sets of integers and real numbers are denoted by $\mathbb{Z}$ and $\mathbb{R}$ respectively. Let $i, j \in \mathbb{Z}$, we denote by $[i, j]$ the set $\{k \in \mathbb{Z} \mid i \leq k \leq j\}$.

We denote by $\mathcal{F}$ the set of all maps from E into $\mathbb{R}$, and we say that any map in $\mathcal{F}$ *weights* the edges of G.

Let $F \in \mathcal{F}$. If u is an edge of G, the number $F(u)$ is the *altitude or weight* of u. Let $k \in \mathbb{R}$, we denote by $F[k]$ the set of edges of G whose weight (for F) is less than or equal to k: $F[k] = \{u \in E \mid F(u) \leq k\}$. The set of edges $F[k]$ is called the *(lower) cross section of F at level k.*

Let $F \in \mathcal{F}$, and let $X \sqsubseteq G$. We say that X is a *connected component of F*, or simply a *component of F*, if there exists an element $k \in \mathbb{R}$ such that X is a component of the graph induced by $F[k]$. We denote by $\mathcal{CC}(F)$ the set of all components of F. The set $\mathcal{CC}(F)$ is closely related to the component tree [28, 25] of F, a tree widely used for filtering.

Let $F \in \mathcal{F}$, and let $X \sqsubseteq G$. The graph X is a *minimum of F* if X is a component of F that does not strictly contains any component of F, *i.e.*, for any $Y \in \mathcal{CC}(F)$, if $Y \sqsubseteq X$, then we have necessarily $Y = X$. We denote by $\mathcal{M}_F$ the set made of the minima of F, and by M_F its supremum: $M_F = \sqcup \mathcal{M}_F$. This notion of minima, as well as further presented notions, is illustrated in Fig. 1

Important notation. *In the sequel of this paper, F denotes an element of $\mathcal{F}$. Therefore the pair (G, F) is called an* edge-weighted graph.

3 Minimum Spanning Forests Hierarchies

This section first presents the minimum spanning forests rooted in subgraphs of G. This notion of a forest, which is useful for (seeded) image segmentation (see *e.g.* [9, 10, 6]), is known to be equivalent to the one of minimum spanning tree studied in combinatorial optimization. Then, hierarchies of minimum spanning forests are introduced. Each such hierarchy induces a hierarchy of partitions on V, which is thus optimal in the sense of rooted minimum spanning forests.

Let $X \sqsubseteq G$. *The weight of X (for F)*, denoted by $F(X)$, is the sum of the weights of the edges in $E(X)$: $F(X) = \sum_{u \in E(X)} F(u)$. Let $V' \subseteq V$, we say that X is spanning for V' if $V(X) = V'$.

Definition 1 (rooted MSF). *Let X and Y be two nonempty subgraphs of G.*

We say that Y is rooted in *X if $V(X) \subseteq V(Y)$ and if the vertex set of any component of Y contains the vertex set of exactly one component of X.*

We say that Y is a minimum spanning forest (MSF) rooted in *X* (with respect to *F) if:*

1. *Y is spanning for V;*
2. *Y is rooted in X; and*

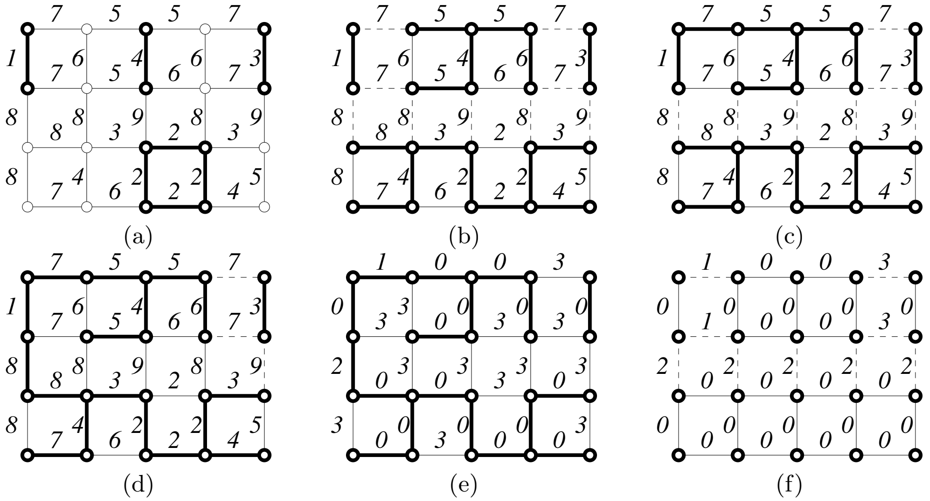


Fig. 1. (a): A graph G and a map F whose minima are depicted in bold. (b): A graph $X_0 \in MSF(M_F)$ represented in bold and its induced cut $C_0 = C(X_0)$ represented by dashed edges. (c,d): two bold graphs called respectively X_1 and X_2 such that $\mathcal{T} = \langle X_0, X_1, X_2 \rangle$ is both an MSF hierarchy for and an uprooting by $\langle M_1, M_2 \rangle$ (where M_i is the minimum of F whose altitude is i); their induced cuts $C_1 = C(X_1)$ and $C_2 = C(X_2)$ are represented by dashed edges. (e): The indicator of the uprooting $\mathcal{U}$. (f): The saliency map of the MSF cut hierarchy $\langle C_0, C_1, C_2 \rangle$.

3. *the weight of Y is less than or equal to the weight of any graph Z satisfying (1) and (2) (*i.e.*, Z is both spanning for V and rooted in X).*

The set of all minimum spanning forests rooted in X is denoted by $MSF(X)$.

The above definition of rooted MSFs, which is illustrated in Fig. 1b, allows the usual notions of graph theory based on trees and forests to be recovered. In particular, if x is a vertex of V, it can be seen that any element in $MSF((\{x\}, \emptyset))$ is a *minimum spanning tree of* (G, F), and that, conversely, any minimum spanning tree of (G, F) belongs to $MSF((\{x\}, \emptyset))$. In the following, by convention, this remarkable set, which is made of all minimum spanning trees of (G, F), will also be denoted by $MSF((\emptyset, \emptyset))$.

A possible definition for watershed, called watershed cuts, follows the drop of water principle. In [9], we have proved the equivalence between MSF rooted in the set of minima and watershed cuts. In practice, watersheds from markers are often computed, and subsets of minima of the original edge-weighted graph constitute robust markers [3]. The next definition, illustrated in Figs. 1b, c, and d, presents a notion of hierarchy of MSFs rooted in such subsets.

Definition 2 (MSF hierarchy). *Let $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ be a sequence of pairwise distinct minima of F and let $\mathcal{T} = \langle X_0, \ldots X_\ell \rangle$ be a sequence of subgraphs of G. We say that $\mathcal{T}$ is an* MSF hierarchy for $\mathcal{S}$ *if:*

1. *for any* $i \in [0, \ell]$*, the graph* X_i *is an MSF rooted in* $\sqcup[\mathcal{M}_F \setminus \{M_j \mid j \in [1, i]\}]$*; and*
2. *for any* $i \in [1, \ell]$*, we have* $X_{i-1} \sqsubseteq X_i$*.*

4 Uprootings and MSF Hierarchies

In this section, we formalize a simple transformation, called uprooting, that allows a forest X rooted in a graph M to be incrementally transformed into a forest Y rooted in a graph M' obtained by removing some components of M. Through an equivalence theorem, we establish an important link between the uprooting transform and the MSF hierarchies. This result allows efficient algorithms for computing MSF hierarchies to be considered.

Let X be a subgraph of G that is spanning for V, and let $x \in V$. We denote by $CC_x(X)$ the component of X whose vertex set contains x. Let $V' \subseteq V$, we set $CC_{V'}(X) = \sqcup\{CC_x(X) \mid x \in V'\}$.

Let $X \subseteq G$, and let $u = \{x, y\} \in E$. The edge u is *outgoing from* X if u is made of a vertex in $V(X)$ and of a vertex in $\overline{V(X)}$. In the following, by abuse of notation, we write $X \sqcup \{u\}$ for the supremum of X and the graph induced by $\{u\}$: $X \sqcup \{u\} = (V(X) \cup u, E(X) \cup \{u\})$.

Let X, Y, and M be three subgraphs of G such that X is spanning for V and such that $X \neq Y$. We say that Y is an *elementary uprooting of* X *by* M if there exists an edge u of minimum weight among the edges outgoing from $CC_{V(M)}(X)$ such that $Y = X \sqcup \{u\}$. We also say that Y is an *elementary uprooting of* X *by* M if $Y = X$ and if there is no edge outgoing from $CC_{V(M)}(X)$.

Definition 3. *Let* $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ *be a sequence of pairwise distinct minima of* F*. An* uprooting by $\mathcal{S}$ *is a sequence* $\langle X_0, \ldots, X_\ell \rangle$ *of graphs such that:*

1. $X_0 \in MSF(M_F)$ *; and*
2. X_i *is an elementary uprooting of* X_{i-1} *by* M_i*, for any* $i \in [1, \ell]$*.*

The following theorem, one of the main results of this paper, states an equivalence property between MSF hierarchies and uprootings by sequences of minima of the original map that weight the edges of the graph.

Theorem 4. *Let* $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ *be a sequence of pairwise distinct minima of* F*, and let* $\mathcal{T} = \langle X_0, \ldots X_\ell \rangle$ *be a sequence of subgraphs of* G*. The sequence* $\mathcal{T}$ *is an MSF hierarchy for* $\mathcal{S}$ *if and only if the sequence* $\mathcal{T}$ *is an uprooting by* $\mathcal{S}$*.*

Any MSF hierarchy can be represented by a unique edge-weighed graph such that a given threshold of this graph provides the associated level of the hierarchy. More formally, let $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ be a sequence of pairwise distinct minima of F, and let $\mathcal{T} = \langle X_0, \ldots, X_\ell \rangle$ be an uprooting by $\mathcal{S}$. The *indicator of* $\mathcal{T}$, denoted by $I_\mathcal{T}$, is the map from E into $[0, \ell + 1]$ defined by:

- $I_\mathcal{T}(u) = \min\{i \in [0, \ell] \mid u \in E(X_i)\}$, for any $u \in E(X_\ell)$; and
- $I_\mathcal{T}(u) = \ell + 1$, for any $u \in E \setminus E(X_\ell)$.

The notion of indicator of an uprooting is illustrated in Fig. 1e.

Any uprooting has a unique indicator, and any two distinct uprootings have distinct indicators. Hence, the indicator of an uprooting is sufficient to recover all the elements of the uprooting: if $\mathcal{T} = \langle X_0, \ldots, X_\ell \rangle$ is an uprooting by a sequence $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ of minima of F, then X_i is the graph induced by the cross-section of $I_\mathcal{T}$ at level i, (*i.e.*, $X_i = (\delta^\bullet(I_\mathcal{T}[i]), I_\mathcal{T}[i])$.

We are now ready to describe the uprooting algorithm, which by Theorem 4 allows the computation of MSF hierarchies. It inputs a sequence $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ of minima of F and outputs the indicator $I_\mathcal{T}$ of an uprooting $\mathcal{T} = \langle X_0, \ldots, X_\ell \rangle$ by $\mathcal{S}$. From a high level point of view, the algorithm can be sketched, once a minimum spanning forest X relative to M has been computed, by iterations of the following three region merging steps:

i Find the component C of X that contains M_i the next element in $\mathcal{S}$.
ii Find a component C' of X linked to C by an edge v of minimum weight
iii Merge the two components C and C'

Algorithm. Uprooting

```
Data: $(V, E, F)$: an edge-weighted graph;
$X \leftarrow$ any element in $MSF(M_F)$;
A sequence $\mathcal{S} = \langle M_0, \ldots, M_\ell \rangle$ of pairwise distinct minima of $F$;
Result: $I$, the indicator of an uprooting by $\mathcal{S}$;
$i \leftarrow 0$; ;                                              /* Initialisation */
foreach $x \in E$ do
    MakeSet($x$); $\mathcal{L}[x] \leftarrow$ MakeList();
foreach $u = \{x, y\} \in E$ do
    if $u \in E(X)$ then $I(u) \leftarrow 0$;else $I(u) \leftarrow \ell + 1$;
foreach $u = \{x, y\} \in E(X)$ do
    $x' \leftarrow$ Find(x); $y' \leftarrow$ Find(y);
    if $x' \neq y'$ then Link($x'$,$y'$);
foreach $u = \{x, y\} \in E \setminus E(X)$ do
    $x' \leftarrow$ Find(x); $y' \leftarrow$ Find(y);
    if $x' \neq y'$ then Insert$_{x'}$($y'$, $u$, $F(u)$); Insert$_{y'}$($x'$, $u$, $F(u)$);
while $i < \ell$ do                                     /* Incremental uprooting */
    $i \leftarrow i + 1$;
    Set $x$ to any element of $V(M_i)$; $x' \leftarrow$ Find($x$);
    repeat
        $(y, v, w) \leftarrow$ DeleteMin$_{x'}$();
        $y' \leftarrow$ Find($y$);
    until $x' \neq y'$ ;
    $I(v) \leftarrow i$;
    $z \leftarrow$ Link($x'$,$y'$); $\mathcal{L}[z] \leftarrow$ Meld($x'$,$y'$);
```

In order to ease the reading of the algorithm, let us first present the two main data structures that are used.

The structure $\mathcal{C}$ is a collection of disjoint subsets of V. Each set C in $\mathcal{C}$ is represented by a unique element x of C, called the canonical element of C. In the following, x and y denote two distinct elements of V. The collection $\mathcal{C}$ is managed by three operations:

- **MakeSet(x):** add the set $\{x\}$ to the collection $\mathcal{C}$, provided that the element x does not already belong to a set in $\mathcal{C}$.
- **Find(x):** returns the canonical element of the set in $\mathcal{C}$ that contains x.
- **Link(x, y):** let C_x and C_y denote two sets in $\mathcal{C}$ whose canonical elements are x and y respectively. Both sets are removed from $\mathcal{C}$, their union $C = C_x \cup C_y$ is added to $\mathcal{C}$ and a canonical element for C is selected and returned.

The structure $\mathcal{L}$ is a collection of $n = |V|$ lists: $\mathcal{L}[x_1], \ldots, \mathcal{L}[x_n]$, with $V = \{x_1, \ldots, x_n\}$. Each element of these lists is a triplet (x, u, w) such that x is a vertex of G, u is an edge of G, and w is an element in $\mathbb{R}$, called the weight of the triplet. In the following, x and y denote two distinct elements of V, u denotes an edge in E, and w an element in $\mathbb{R}$. The collection $\mathcal{L}$ is managed by four operations:

- **MakeList():** returns an empty list.
- **Insert$_x$(y, u, w):** adds the triplet (y, u, w) to the list $\mathcal{L}[x]$;
- **DeleteMin$_x$()** returns and removes from $\mathcal{L}[x]$ a triplet of minimum weight.
- **Meld(x, y):** returns the list of the triplets of $\mathcal{L}[x]$ and of $\mathcal{L}[y]$ and suppresses the lists $\mathcal{L}[x]$ and $\mathcal{L}[y]$.

When the algorithm terminates the map I is the indicator of an uprooting by the input sequence $\mathcal{S}$. Moreover, using Tarjan's union find [29] and Fredman and Tarjan's Fibonnacci heap [14] algorithms to manage the collections $\mathcal{C}$ and $\mathcal{L}$, the overall complexity of the algorithm is $O(|V| \times \alpha(|V|, |V|) + |E| \times \alpha(|E|, |V|) + |E| \log |E|)$, where α is a function which grows very slowly: for all practical purposes $\alpha(m, n)$ is never greater than four. In other words, the complexity of the algorithm is quasi $O(|V| + |E| \log |E|)$ in the sense of Tarjan's union find.

5 MSF Cut Hierarchy, Saliency and Connection Value

Until now, we have dealt with regions (components of forests). Let us now study their "dual", called cuts, that represent borders between regions.

Let X be a subgraph of G that is spanning for V. The *(graph) cut induced by* X, denoted by $C(X)$, is the set of all edges of G made of vertices of two distinct components of X, *i.e.* $C(X) = \{\{x, y\} \in V \mid CC_x(X) \neq CC_y(X)\}$.

Definition 5 (MSF cut hierarchy). *Let X be a subgraph of G, and let $C \subseteq E$. The set C is called an* MSF cut for X (with respect to F) *if there exists an MSF Y rooted in X, such that C is the cut induced by Y.*

Let $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ be a sequence of pairwise distinct minima of F and let $\mathcal{T} = \langle C_0, \ldots C_\ell \rangle$ be a sequence of subsets of E. We say that $\mathcal{T}$ is an MSF cut hierarchy *for $\mathcal{S}$ if:*

1. *for any* $i \in [0, \ell]$*, the set* C_i *is an MSF cut for* $\sqcup[\mathcal{M}_F \setminus \{M_j \mid j \in [1, i]\}]$*; and*
2. *for any* $i \in [1, \ell]$*, we have* $C_i \subseteq C_{i-1}$*.*

The following result asserts that there is indeed an equivalence relation between MSF hierarchies and MSF cut hierarchies.

Property 6. *Let* $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ *be a sequence of pairwise distinct minima of* F *and let* $\mathcal{T} = \langle C_0, \ldots C_\ell \rangle$ *be a sequence of subsets of* E*. The sequence* $\mathcal{T}$ *is an MSF cut hierarchy if and only if there exists an MSF hierarchy* $\langle X_0, \ldots, X_\ell \rangle$ *such that* $C(X_i) = C_i$*, for any* $i \in [0, \ell]$*.*

The hierarchies presented above are hierarchical segmentations as defined in [24]. Therefore, the MSF cuts belonging to a hierarchy can be "stacked" to form a map that equivalently represents this hierarchy. Intuitively, such a map, called a saliency map, weights the cuts with their "level of disappearance" in the hierarchy. Hence, they are convenient for visualizing hierarchies.

Definition 7 (saliency map). *Let* $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ *be a sequence of minima of* F *and let* $\mathcal{T} = \langle C_0, \ldots C_\ell \rangle$ *be an MSF cuts hierarchy for* $\mathcal{S}$*. The* saliency map of $\mathcal{T}$*, denoted by* $S_\mathcal{T}$*, is the map from* E *into* $[0, \ell + 1]$ *defined by:*

1. $S_\mathcal{T}(u) = \min\{i \in [0, \ell] \mid u \in \overline{C_i}\}$*, for any* $u \in E \setminus C_\ell$*; and*
2. $S_\mathcal{T}(u) = \ell + 1$*, for any* $u \in C_\ell$*.*

An illustration of the notion of saliency map is given in Fig. 1f.

Let H be any map and let $k \in \mathbb{R}$, the set $\overline{H[k]}$ of the edges of G whose weight (for H) is greater than k is called the *upper cross section of* H *at level* k*.*

Let $\mathcal{T} = \langle C_0, \ldots, C_\ell \rangle$ be an MSF cut hierarchy for a sequence $\mathcal{S}$ of minima of F, and let H be a map from E into $[0, \ell + 1]$. It can be seen that $H = S_\mathcal{T}$ if and only if, for any $i \in [0, \ell]$, the set C_i is the upper cross-section of H at level i. Thus, the saliency map $S_\mathcal{T}$ is an equivalent representation of the hierarchy $\mathcal{T}$.

The altitudes of the passes between minima of the image play a fundamental role in morphological filtering. In our framework, pass altitudes are called connection values. In the following, we show their importance for computing a saliency map from the indicator of an uprooting.

Let $H \in \mathcal{F}$ and let $X \subseteq G$. The altitude of X for H, denoted by $\check{H}(X)$, is the maximum value of an edge of X.

Definition 8 (connection value). *Let* $H \in \mathcal{F}$*, and let* x *and* y *be two points of* G*. The* connection value between x and y for H (in G) *is the value* $\Upsilon_H(x, y) = \min\{\check{H}(X) \mid X \in \mathcal{CC}(F)$ *and* $\{x, y\} \subseteq V(X)\}$*.*

The connection value, for any map H, between any two points can be computed in constant time (see [7]) from the (min) component tree $(\mathcal{CC}(H), \sqsubseteq)$ of H, provided a linear time preprocessing (see [4]). In other words, once the component tree is built (see [25] for a quasi linear time algorithm) and preprocessed (in linear time), computing the connection value between any arbitrary pair of points can be done in constant time.

Hierarchical segmentations are in bijection with ultrametric watersheds [24]. The following theorem provides an analog result for MSF cut hierarchies.

Theorem 9. *Let $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ be a sequence of pairwise distinct minima of F, and let H be a map from E into $[0, \ell + 1]$. The map H is the saliency map of an MSF cut hierarchy for $\mathcal{S}$ if and only if there exists an uprooting $\mathcal{T}$ by $\mathcal{S}$ such that, for any $u = \{x, y\}$, $F(u) = \Upsilon_{I_\mathcal{T}}(x, y)$.*

Hence, in order to know the value of an edge in a saliency map it is sufficient to know the connection value between the two points linked by this edge for the indicator of an uprooting by $\mathcal{S}$. Thus, knowing the indicator of an uprooting by $\mathcal{S}$ and its component tree, a saliency map can be computed in linear time, *i.e.*, one constant time operation per edge of G.

Figs. 2b,c, and d illustrates the use of this algorithm on the image of Fig. 2a. The underlying graph is the one induced by the 4-adjacency relation whose edges are weighted by a simple color gradient (maximum, over the RGB channels,

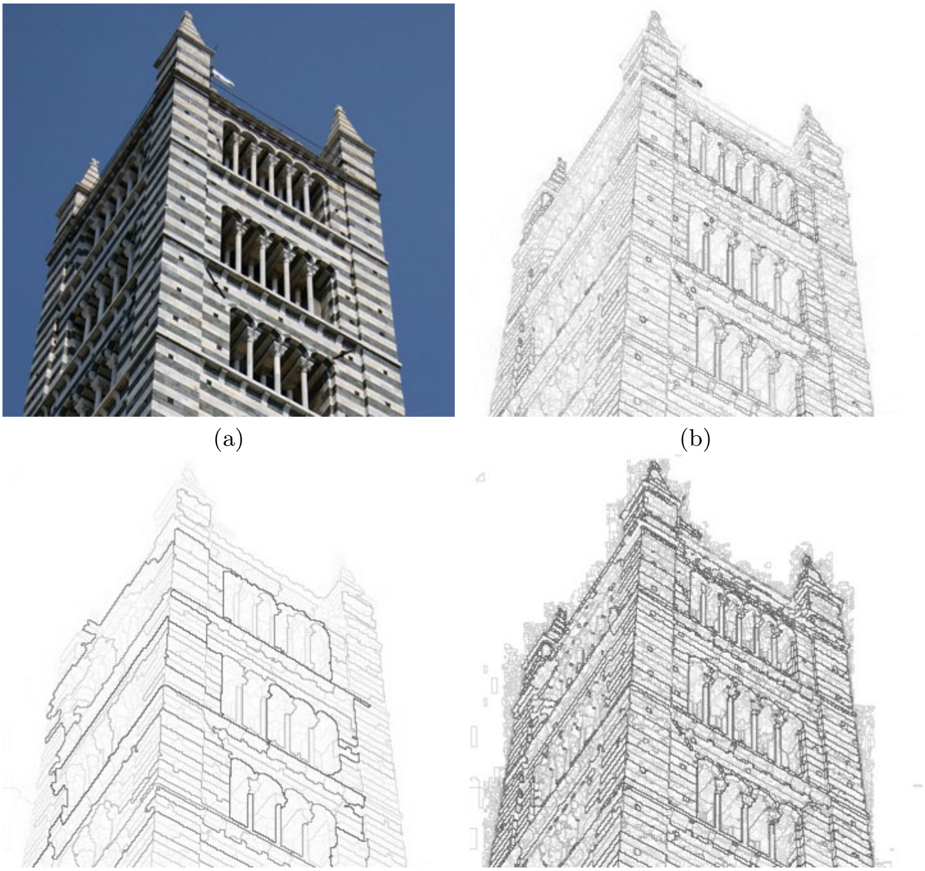

(a)
(b)
(c)
(d)

Fig. 2. Illustration of saliencies of watershed cuts (original picture (a) from *koakoo*: http://blog.photos-libres.fr/)

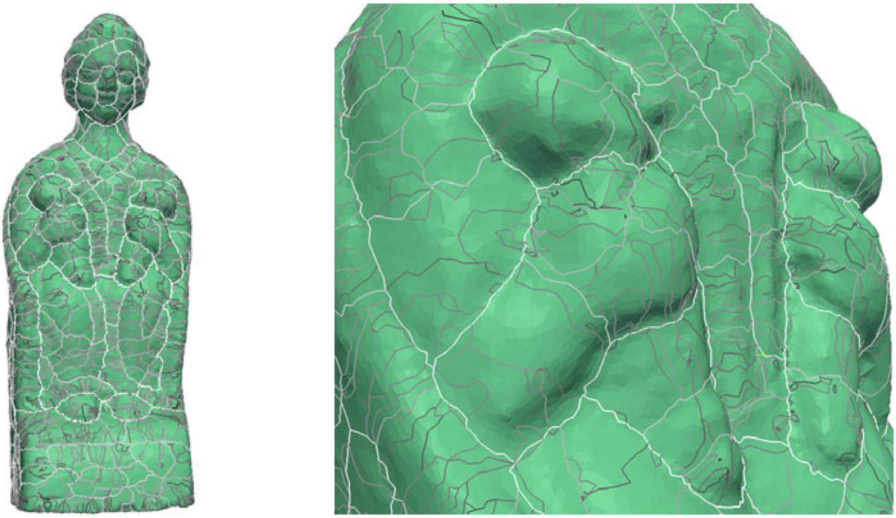

(a) (b): zoom on a part of (a)

Fig. 3. Illustration of saliencies of watershed cuts on a mesh provided by the French Museum Center for Research and Restoration (C2RMF, Le Louvre, Paris)

of the absolute differences of pixel values). The minima are ordered thanks to extinction values [30] related to dynamics, volume and color consistency, leading to sequences $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ of minima. The saliency maps of MSF hierarchies for $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ are rendered (up to an anamorphosis) in sub-figures b,c, and d, respectively. Figs. 3 illustrates the use of the algorithm to segment a surface (*i.e.*, a mesh) embedded in the 3D Euclidean space. The vertex set of the considered graph is the set of triangles of the mesh, and its edge set is composed by the pair of triangles that share a common side. The edges of this graph are weighted by map which behaves like the inverse of the mean curvature of the surface (see [27]). The minima are ordered thanks to volume extinction values and the saliency maps of the induced MSF hierarchies is rendered up to an anamorphosis.

6 Hierarchy by Geodesic Reconstructions

A desirable compatibility property in the context of morphological filtering is that any threshold of a saliency map is a watershed of the geodesic reconstruction of the original map by the minima corresponding to the threshold level. Theorem 11 below shows that the results of our algorithm do satisfy such a compatibility property. It has to be noted that, in other frameworks (*e.g.*, node-weighted graphs), such a property is in general not guaranteed [26].

Let $x \in V$. We denote by $\mathcal{CC}_x(F)$ the set of all components of F whose vertex sets contain x. Let $V' \subseteq V$, we set $\mathcal{CC}_{V'}(F) = \cup\{\mathcal{CC}_x(F) \mid x \in V'\}$.

Let $V' \subseteq V$. The *geodesic reconstruction of F by V'* is the map F' such that, for any edge u in E, the value $F'(u)$ is the minimum altitude of a component in $\mathcal{CC}_{V'}(F)$ that contains u: $F'(u) = \min\{\check{F}(X) \mid X \in \mathcal{CC}_{V'}(F)\}$.

Watershed from markers are classically defined through geodesic reconstruction (also called swamping or flooding). It is shown in [1] that reconstruction plus MSF cut can be replaced by an MSF cut relative to the initial map.

Property 10 (from Theorem 6.3 in [1]). *Let $V' \subseteq V$, let F' be the geodesic reconstruction of F by V', and let C be a subset of E. If C is an MSF cut for M_H with respect to F, then C is an MSF cut for M_H with respect to F'.*

The converse of Prop. 10 is, in general, not true. In fact, as long as a sequence of nested partitions (*i.e.*, a hierarchy) is involved, the MSF cuts of reconstructed maps and the MSF cuts of the original maps are equivalent.

Theorem 11 (Compatibility). *Let $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ be a sequence of pairwise distinct minima of F. Let G_i denote $\sqcup[\mathcal{M}_F \setminus \{M_j \mid j \in [1,i]\}]$, and let F_i be the geodesic reconstruction of F by $V(G_i)$, for any $i \in [0,\ell]$. Let $\mathcal{T} = \langle C_0, \ldots, C_\ell \rangle$ be a sequence of subsets of E such that $C_i \subseteq C_{i-1}$ for any $i \in [1,\ell]$. Then, the two following propositions are equivalent:*

1. *for any $i \in [0,\ell]$, the set C_i is an MSF cut for G_i with respect to F_i; and*
2. *for any $i \in [0,\ell]$, the set C_i is an MSF cut for G_i with respect to F.*

7 Conclusion and Perspective

In this paper, a classical morphological scheme for building hierarchical segmentation is formalized. This formalism leads us to establish strong properties linking hierarchical segmentations and combinatorial optimality in terms of minimum spanning forests of the original image. Recent work of Couprie *et al.* [6] link some schemes based on minimum spanning forests to global energy minimization. Hence, a promising perspective is the investigation of hierarchical schemes defined through energy minimization such as the one presented in [16].

The source code of the algorithms presented in this paper is available at: `http://www.esiee.fr/~info/sm`

References

1. Allène, C., Audibert, J.-Y., Couprie, M., Keriven, R.: Some links between extremum spanning forests, watersheds and min-cuts. IVC 28(10), 1460–1471 (2010)
2. Arbeláez, P.A., Cohen, L.D.: A metric approach to vector-valued image segmentation. IJCV 69(1), 119–126 (2006)
3. Audigier, R., Lotufo, R.: Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches. In: IEEE SIBGRAPI 2007, pp. 61–70 (2007)
4. Bender, M.A., Farach-Colton, M.: The LCA problem revisited. In: Latin American Theoretical Informatics, pp. 88–94 (2000)
5. Beucher, S.: Watershed, hierarchical segmentation and waterfall algorithm. In: ISMM 1994, pp. 69–76 (1994)
6. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power Watersheds: A Unifying Graph Based Optimization Framework. PAMI (to appear, 2011)

7. Couprie, M., Najman, L., Bertrand, G.: Quasi-linear algorithms for the topological watershed. JMIV 22(2-3), 231–249 (2005)
8. Cousty, J., Najman, L., Serra, J.: Raising in watershed lattices. In: 15th IEEE ICIP 2008, pp. 2196–2199 (2008)
9. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle. PAMI 31(8), 1362–1374 (2009)
10. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: thinnings, shortest-path forests and topological watersheds. PAMI 32(5), 925–939 (2010)
11. Cousty, J., Najman, L., Bertrand, G., Couprie, M.: Weighted fusion graphs: merging properties and watersheds. DAM 156(15), 3011–3027 (2008)
12. Cousty, J., Najman, L., Serra, J.: Some morphological operators in graph spaces. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 149–160. Springer, Heidelberg (2009)
13. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. International Journal of Computer Vision 59, 167–181 (2004)
14. Fredman, M.L., Tarjan, R.E.: Fibonacci heaps and their uses in improved network optimization algorithms. J. ACM 34, 596–615 (1987)
15. Gower, J.C., Ross, G.J.S.: Minimum spanning tree and single linkage cluster analysis. Appl. Stats. 18, 54–64 (1969)
16. Guigues, L., Cocquerez, J.P., Men, H.L.: Scale-sets image analysis. IJCV 68(3), 289–317 (2006)
17. Jardine, N., Sibson, R.: Mathematical taxonomy. Wiley, Chichester (1971)
18. Marcotegui, B., Beucher, S.: Fast implementation of waterfall based on graphs. In: ISMM 2005, pp. 177–186 (2005)
19. Meyer, F.: Minimum spanning forests for morphological segmentation. In: ISMM 1994, pp. 77–84 (1994)
20. Meyer, F.: The dynamics of minima and contours. In: ISMM, pp. 329–336 (1996)
21. Meyer, F., Najman, L.: Segmentation, minimum spanning tree and hierarchies. In: Mathematical Morphology, ch. 9, pp. 229–261. ISTE-Wiley (2010)
22. Morris, O.J., Lee, M.d.J., Constantinides, A.G.: Graph theory for image analysis: an approach based on the shortest spanning tree. IEE Proc. on Communications, Radar and Signal 133(2), 146–152 (1986)
23. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. PAMI 18(12), 1163–1173 (1996)
24. Najman, L.: On the equivalence between hierarchical segmentations and ultrametric watersheds. JMIV 40(3), 231–247 (2011)
25. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE TIP 15(11), 3531–3539 (2006)
26. Najman, L., Couprie, M., Bertrand, G.: Watersheds, mosaics and the emergence paradigm. DAM 147(2-3), 301–324 (2005)
27. Philipp-Foliguet, S., Jordan, M., Najman, L., Cousty, J.: Artwork 3D Model Database Indexing and Classification. Patt. Recogn. 44(3), 588–597 (2011)
28. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE TIP 7(4), 555–570 (1998)
29. Tarjan, R.E.: Efficiency of a good but not linear set union algorithm. J. ACM 22, 215–225 (1975)
30. Vachier, C., Meyer, F.: Extinction value: a new measurement of persistence. In: IEEE Workshop on Nonlinear Signal and Image Processing, pp. 254–257 (1995)
31. Zahn, C.T.: Graph-theoretical methods for detecting and descibing gestalt clusters. IEEE Transactions on Computers C-20(1), 99–112 (1971)

Component-Hypertrees for Image Segmentation

Nicolas Passat[1,⋆] and Benoît Naegel[2]

[1] Université de Strasbourg, LSIIT, UMR CNRS 7005, France
[2] Université Nancy 1, LORIA, UMR CNRS 7503, France
passat@unistra.fr, benoit.naegel@loria.fr

Abstract. This article introduces the notion of *component-hypertree*, which models the component-trees of an image at various connectivity levels, and the relations of the nodes/connected components between these levels. This data structure is then used to extend a recently proposed interactive segmentation method based on component-trees. In this multiscale connectivity context, the use of a component-hypertree appears to be less costly than the use of several component-trees. Application examples illustrate the relevance of this approach.

Keywords: Component-tree, segmentation, mask-based connectivity, hypertree.

1 Introduction

The component-tree [13] is a graph-based structure which models some characteristics of a grey-level image by considering its binary level-sets obtained from successive thresholdings. It is particularly well-suited for the design of fast segmentation methods [5,10], based on hypotheses related to the topology (connectedness) and the specific intensity (locally extremal) of structures of interest.

In this context, a new segmentation method has been proposed recently by Passat *et al.* [12]. For a given (manual) presegmentation, it computes the best segmentation induced by the nodes of the image component-tree. By opposition to the standard component-tree approaches, based on attributes [16], this "best segmentation" is here the one which minimises a cost function modelling false positives/negatives between the manual approximate segmentation and the proposed solution.

This method provides results for a given connectivity (in general, a connectivity induced by the standard adjacencies on $\mathbb{Z}^n$). In order to also take into account simultaneously several (richer) connectivities, and in particular mask-based connectivities [11], an extension of this method is proposed. Based on a *component-hypertree*, which models the component-trees of a same image at various connectivity levels (induced by increasing masks), and the decomposition of each node/connected component in these successive component-trees, it computes all the segmentation results induced by these connectivity levels.

⋆ The research leading to these results has received funding from the French *Agence Nationale de la Recherche* (Grant Agreement ANR-2010-BLAN-0205).

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 284–295, 2011.

This extension enables, in particular, to minimise the algorithmic time cost, by comparison with a straightforward naive approach.

The article is organised as follows. In Section 2, a (short) state of the art related to component-tree segmentation is described. In Section 3, the segmentation method defined by Passat *et al.* [12] is recalled. Section 4, which contains the contribution of this article, presents the notion of component-hypertree, which provides a way to handle multiscale connectivity. It also describes how the method described in Section 3 can be interfaced with component-hypertrees to enable the segmentation of an image at various connectivity levels, in an efficient algorithmic fashion. In Section 5, experiments illustrate the behaviour of this technique. Conclusions can be found in Section 6.

2 Previous Work

Component-trees have been involved in the development of several applications related to image filtering and/or segmentation [11,19,18,8,1,9]. All these proposed methods have been designed to detect the structures of interest by using information related to the value of attributes stored at each node of the tree. In such strategies, an attribute, or more generally a set of attributes [16], are chosen according to the hypotheses related to the applicative context. These attributes are assumed to model some characteristic properties of the structures of interest, and can be used in different ways:

- the desired values of the attributes can be chosen by the user in order to select the relevant nodes inducing the correct segmentation [11,19,18,1];
- these values can be determined by analysing the signature of the attributes (*i.e.*, their evolution with respect to the grey-level of the nodes) [5], possibly in an automatic fashion [6];
- they can be learnt from examples, *e.g.*, by providing a ground-truth characterising the shape of the objects to detect [8], or by feeding a classification process when the set of attributes becomes too large [9,17,4].

In such works, component-trees have been used for their ability to discriminate nodes with respect to attributes, leading to automated/parametric methods.

It is however possible to directly use the component-tree structure by taking advantage of the decomposition of the image into nodes/connected components that it provides, in order to perform interactive segmentation. Methods based on such an alternative strategy should compute a segmentation result, no longer thanks to node attributes, but to a user-defined approximate result, which should then be matched at best by a relevant set of nodes. To our knowledge, the methodology proposed in the next sections is the first one based on this strategy.

3 Component-Tree Segmentation Method

3.1 Component-Trees

Let $E \subset \mathbb{Z}^n$ ($n \geq 1$) be a finite connected set (for a given adjacency relation). Let $V = [\![\perp, \top]\!] \cup \{-\infty\}$ with $[\![\perp, \top]\!] \subset \mathbb{Z}$. A grey-level image is a function

$I : \mathbb{Z}^n \to V$ such that $I^{-1}(\{-\infty\}) = \mathbb{Z}^n \setminus E$. By abuse of notation, we will implicitly restrict I to E, and note $I : E \to V$, or $I \in V^E$. We assume, without loss of generality, that $\top = \max\{I(x) \mid x \in E\}$.

The thresholding function $X_v : V^E \to 2^E$ ($v \in V$) is defined by $X_v(I) = \{x \in E \mid v \leq I(x)\}$. The cylinder function $C_{X,v} : E \to V$ ($v \in V$, $X \subseteq E$) is defined by $C_{X,v}(x) = v$ if $x \in X$ and $-\infty$ otherwise. An image $I \in V^E$ can be written as $I = \bigvee_{v\in V} \bigvee_{X\in\mathcal{C}[X_v(I)]} C_{X,v}$ where $\bigvee$ is the pointwise supremum for the sets of functions, and $\mathcal{C}[\ .\ ]$ is the set of the connected components of a set.

Let $\mathcal{K} = \bigcup_{v\in V} \mathcal{C}[X_v(I)]$ be the set of the connected components generated by the thresholdings of I at all values $v \in V$. The component-tree of I is obtained from the Hasse diagram of the partially ordered set $(\mathcal{K}, \subseteq)$.

Definition 1 (Component-tree). *Let $I \in V^E$ be a grey-level image. The* component-tree *of I is the rooted tree $T = (\mathcal{K}, L, R)$ such that:*

(i) $\mathcal{K} = \bigcup_{v\in V} \mathcal{C}[X_v(I)]$;
(ii) $L = \{(X,Y) \in \mathcal{K}^2 \mid Y \subset X \wedge \forall Z \in \mathcal{K}, Y \subseteq Z \subset X \Rightarrow Y = Z\}$;
(iii) $R = \sup(\mathcal{K}, \subseteq) = X_\perp(I) = E$.

The elements of $\mathcal{K}$ (resp. of L) are the nodes *(resp. the* edges*) of T. The node R is the* root *of T. For any $N \in \mathcal{K}$, we set $ch(N) = \{N' \in \mathcal{K} \mid (N, N') \in L\}$; $ch(N)$ is the set of the* children *of N. If $ch(N) = \emptyset$, we say that N is a* leaf.

3.2 Problem to Solve

Component-trees can be used to develop segmentation procedures which consist of determining a subset $\widehat{\mathcal{K}} \subseteq \mathcal{K}$ among the nodes of the component-tree $T = (\mathcal{K}, L, R)$ of an image $I : E \to V$. The binary result $I_s \subseteq E$ is then defined as $I_s = \bigcup_{X\in\widehat{\mathcal{K}}} X$. A way to consider this segmentation problem is to search the set of nodes $\widehat{\mathcal{K}} \subseteq \mathcal{K}$ which enables to generate a binary object being as similar as possible to a target (*e.g.*, a manual presegmentation). This issue can be formalised as the resolution of the following optimisation problem

$$\widehat{\mathcal{K}} = \arg\min_{\mathcal{K}'\subseteq\mathcal{K}} \Big\{ d\Big(\bigcup_{N\in\mathcal{K}'} N, M \Big) \Big\} \tag{1}$$

where $M \subseteq E$ is the (binary) target, and d is a (pseudo-)distance on 2^E. An intuitive solution for determining such a useful pseudo-distance is to consider the number of false positives/negatives induced by $X = \bigcup_{N\in\mathcal{K}'} N$ with respect to M

$$d^\alpha(X, M) = \alpha.|X \setminus M| + (1-\alpha).|M \setminus X| \text{ with } \alpha \in [0,1].$$

3.3 Segmentation Method

As the set $\mathcal{K}$ is finite, there exists a solution to Equation (1). The function $\mathcal{F}^\alpha$ proposed in Definition 2 enables to build a binary image whose connected components form a set $\widehat{\mathcal{K}}$ which is a solution of Equation (1) (see Proposition 3).

Definition 2 ([12]). *Let $\alpha \in [0,1]$. Let $I \in V^E$. Let $T = (\mathcal{K}, L, R)$ be the component-tree of I. Let $M \subseteq E$. Let $\prec \in \{<, \leq\}$. Let $\mathcal{F}^\alpha : \mathcal{K} \to 2^{\mathcal{K}}$ and $c^\alpha : \mathcal{K} \to \mathbb{R}^+$ be the functions recursively cross-defined, for all $N \in \mathcal{K}$, by*

$$\begin{cases} \mathcal{F}^\alpha(N) = \{N\} \\ c^\alpha(N) = \alpha.n(N, M) \end{cases}$$

if

$$\alpha.n(N, M) \prec (1-\alpha).p^*(N, M) + \sum_{N' \in ch(N)} c^\alpha(N') \tag{2}$$

and

$$\begin{cases} \mathcal{F}^\alpha(N) = \bigcup_{N' \in ch(N)} \mathcal{F}^\alpha(N') \\ c^\alpha(N) = (1-\alpha).p^*(N, M) + \sum_{N' \in ch(N)} c^\alpha(N') \end{cases}$$

otherwise, where $p^*(N, M) = |(N \setminus \bigcup_{N' \in ch(N)} N') \cap M|$, and $n(N, M) = |N \setminus M|$.

Proposition 3 ([12]). *We set $M^\alpha = \bigcup_{N \in \mathcal{F}^\alpha(E)} N$. Then, we have*

$$d^\alpha(M^\alpha, M) = c^\alpha(E) = \min_{\mathcal{K}' \subseteq \mathcal{K}} \left\{ d^\alpha\big(\bigcup_{N \in \mathcal{K}'} N, M \big) \right\} .$$

Proposition 4 ([12]). *$\mathcal{F}^\alpha(E) = \mathcal{C}[M^\alpha]$ (and thus M^α) can be computed with the linear algorithmic complexity $\mathcal{O}(\max\{|\mathcal{K}|, |E|\})$.*

4 Component-Hypertree and Segmentation

Section 4.1 describes the connectivity framework considered in this work. In Section 4.2, the notion of component-hypertree is introduced and discussed. In Section 4.3, the way to model a component-hypertree in a compact fashion is exposed, thus inducing a decrease in space cost. Section 4.4, describes how the component-hypertree can be involved in the segmentation method of Section 3, leading to a multiscale connectivity segmentation method[1]. Finally, Section 4.5 provides some properties which show that this way to introduce the component-hypertree also leads to a decrease in time cost by comparison to the only use of component-trees.

4.1 Mask-Based Connectivity

Two points of $\mathbb{Z}^n$ are generally said to be adjacent if their "continuous analogues" in $\mathbb{R}^n$ have a specific (non-empty) intersection. This leads to the standard notions of $2n$- and $(3^n - 1)$-adjacencies in $\mathbb{Z}^n$ [7]. In this framework, connectivity derives from a notion of path (*i.e.*, a series of successively adjacent points in $\mathbb{Z}^n$) which complies with the notion of path in $\mathbb{R}^n$.

[1] The approach presenting the strongest similarities with this work is probably the one proposed by Soille in [15].

A (morphological) alternative definition has been proposed with the notion of second-generation connectivity [14]. This approach of connectivity has led, in particular, to consider the notion of multiscale connectivity [2,3].

In the context of second-generation connectivity, mask-based connectivity [11] proposes to use some masks in order to characterise connected sets. In the binary case, by only considering masks which are either subsets or supersets of an image, we derive from [11] the following definition.

Definition 5 (Mask-based connectivity). *Let $X \subseteq E$. Let $\omega(X) \subseteq X$ (or $\supseteq X$) be a mask of X. The ω-connected components of X, noted $\mathcal{C}_\omega[X]$, are defined as follows:*

- *in the case $\omega(X) \subseteq X$, the ω-connected components of X are:*
  - *the connected components of $\omega(X)$; and*
  - *the singleton sets $\{x\}$ for any $x \in X \setminus \omega(X)$;*
- *and in the case $\omega(X) \supseteq X$, the ω-connected components of X are:*
  - *the sets $X \cap Y$, for any connected component Y of $\omega(X)$.*

In the sequel, for a given image $I : E \to V$, we consider extensive (resp. anti-extensive) masks $\Omega_\star(I) : E \to V$, *i.e.*, masks verifying $I(x) \leq \Omega_\star(I)(x)$ (resp. $I(x) \geq \Omega_\star(I)(x)$) for all $x \in E$. We call $\Omega_\star$-connected components of I, and we note $\mathcal{K}_\star$ the set of all the $\omega_\star$-connected components of $X_v(I)$ induced by the masks $\omega_\star(X_v(I)) = X_v(\Omega_\star(I))$, at all values $v \in V$.

We consider, in particular, the families of masks $\{\Omega_i(I)\}_{i=t}^{u}$ ($t \leq 0 \leq u$) such that (i) $\Omega_t(I) = C_{E,-\infty}$, ($ii$) $\Omega_0(I) = I$, (iii) $\Omega_u(I) = C_{E,\top}$, and (iv) $\Omega_i(I) < \Omega_j(I)$ for any $t \leq i < j \leq u$. An example of such a family $\{\Omega_i(I)\}_{i=-2}^{2}$ is depicted for a 1-D image, in Figure 1. The mask-based connectivity proposed in Definition 5 enables to generate such families of masks. Note however that, without loss of generality, the sequel of the presented work remains valid for any family of masks satisfying properties (i)–(iv), and in particular those induced by alternative connectivities.

Typical examples of families of masks verifying these properties are those induced by erosions/dilations (resp. openings/closings) (with a structuring element containing $0_{\mathbb{Z}^n}$), *e.g.*:

$$\ldots < \epsilon^k(I) < \ldots < \epsilon(I) < I < \delta(I) < \ldots < \delta^k(I) < \ldots$$

$$\ldots < \delta^k \circ \epsilon^k(I) < \ldots < \delta \circ \epsilon(I) < I < \epsilon \circ \delta(I) < \ldots < \epsilon^k \circ \delta^k(I) < \ldots$$

We can, of course, build the ($\Omega_\star$-)component-tree of I induced by the $\Omega_\star$-connected components of the successive level-sets of I. In particular, it can be observed that any node $N \in \mathcal{K}_i$ of the Ω_i-component-tree of I is partitioned into (one or several) node(s) of the Ω_{i-1}-component-tree of I.

4.2 Component-Hypertrees

We consider the definitions of Section 3.1, with the following modifications. An image I is now considered as defined from $\mathbb{Z}^n$ to V (but still with $E = I^{-1}(V \setminus$

$\{-\infty\}$) connected and finite). We also associate to I a mask image $\Omega_\star(I) : \mathbb{Z}^n \to V$ satisfying the hypotheses of Section 4.1, *i.e.*, verifying either $\Omega_\star(I) \leqslant I$ or $\Omega_\star(I) \geqslant I$. Under such conditions, the $\Omega_\star$-connected components of I enable to generate an $\Omega_\star$-component-tree of I similar to the one described in Definition 1, with the following differences: (i) $\mathbb{Z}^n \in \mathcal{K}_\star$, (ii) $R = \sup(\mathcal{K}_\star, \subseteq) = X_{-\infty}(I) = \mathbb{Z}^n$. All the properties previously stated for component-trees remain, however, valid. (Note that for $\Omega_\star(I) = I$, we retrieve the "standard" component-tree, with only one supplementary node which corresponds to a "super-root".)

Definition 6 (Component-hypertree). *Let $I \in V^{\mathbb{Z}^n}$. Let $\{\Omega_i(I)\}_{i=t}^{u}$ $(t \leq 0 \leq u)$ be a set of mask images of I. The* component-hypertree *of I is the triplet $H = (\mathcal{H}, L_\downarrow, L_\rightarrow)$ such that:*

(i) *$\mathcal{H}$ is the multiset*[2] *defined by: $\mathcal{H} = \bigcup_{i=t}^{u} \mathcal{K}_i$;*

(ii) *$\forall i \in [\![t, u]\!]$, the subgraph of $(\mathcal{H}, L_\downarrow)$ induced by the subset of nodes $\mathcal{K}_i \subseteq \mathcal{H}$ is the Ω_i-component-tree of I;*

(iii) *$\forall v \in V$, the subgraph of $(\mathcal{H}, L_\rightarrow)$ induced by the subset of nodes $\mathcal{S}_v = \bigcup_{i \in [\![t,u]\!]} \mathcal{C}_{\omega_i}[X_v(I)] \subseteq \mathcal{H}$ is the Hasse diagram of the partially ordered (multi)set*[3] *$(\mathcal{S}_v, \subseteq)$. This subgraph can, in particular, be seen as a n-ary partition-tree.*

The component-hypertree related to Figure 1 is illustrated in Figure 2.

Remark 7. *Since $\Omega_t(I) = C_{E,-\infty}$, we have $\mathcal{K}_t = \{\mathbb{Z}^n\} \cup \{\{x\} \mid x \in E\}$,* i.e., *the Ω_t-component-tree is composed of a root $\mathbb{Z}^n$ and $|E|$ leaves corresponding each one to a point in E (see right part of Figure 2). Since $\Omega_u(I) = C_{E,\top}$, we have $\mathcal{K}_u = \{X_v(I)\}_{v \in V}$,* i.e., *the Ω_u-component-tree is composed of the $|V|$ binary images corresponding to the successive thresholdings of I; in particular, each node has exactly one child, except $X_\top(I)$ (see left part of Figure 2).*

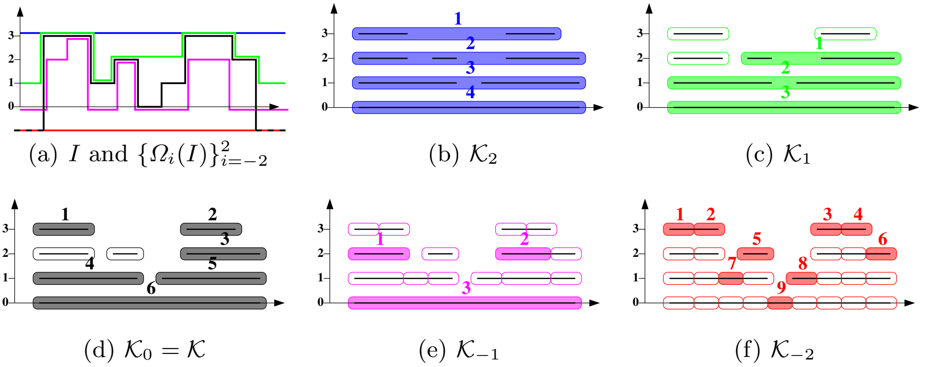
Remark 8. *A node $N \in \mathcal{K}_i$ can have several (distinct) decompositions in $\mathcal{K}_{i-1}$ (this may happen since $\mathcal{K}_i$, is* not *a multiset). In particular, a node N which is a Ω_i-connected component of $X_v(I)$ for all $v \in [\![v^-, v^+]\!] \subseteq V$ can have up to $v^+ - v^- + 1$ decompositions in $\mathcal{K}_{i-1}$, at the same grey-levels. The nodes of these decompositions form a set of (sub)trees of the Ω_{i-1}-component-tree of I (see Figure 2 for a node with two decompositions (red and green edges), and a node with the same decomposition at values* 2 *and* 3 *(magenta edge)).*

4.3 Simplification

Performing segmentation in the way of Section 3 consists of solving Equation (1), *i.e.*, of computing $\mathcal{F}^\alpha(\mathbb{Z}^n)$, for all the $\Omega_\star$-component-trees (in order to allow the user to define, by "connectivity tuning", the most satisfactory result). Instead of

[2] Several nodes of $\mathcal{H}$ which are the same subset of $\mathbb{Z}^n$ can possibly refer to $\Omega_\star$-connected components of distinct $\Omega_\star$-component-trees.

[3] We set $X \subset Y$ whenever $X \in \mathcal{K}_i$, $Y \in \mathcal{K}_j$ are the same subset of $\mathbb{Z}^n$, with $i < j$.



(a) I and $\{\Omega_i(I)\}_{i=-2}^{2}$ (b) $\mathcal{K}_2$ (c) $\mathcal{K}_1$

(d) $\mathcal{K}_0 = \mathcal{K}$ (e) $\mathcal{K}_{-1}$ (f) $\mathcal{K}_{-2}$

Fig. 1. A 1-D image $I : \mathbb{Z} \to [\![0, 3]\!] \cup \{-\infty\}$, and a family $\{\Omega_i\}_{i=-2}^{2}$ of mask images of I. (a) I (in black); $\Omega_2(I) = C_{E,2}$ (in blue); $\Omega_1(I)$ (in green); $\Omega_0(I) = I$ (in black); $\Omega_{-1}(I)$ (in magenta); $\Omega_{-2}(I) = C_{E,-\infty}$ (in red). (b–f) Black lines: connected components of I; black lines inside (filled and unfilled) colour boxes: $\Omega_\star$-connected components of I (from b to f: Ω_2- to Ω_{-2}-connected components).

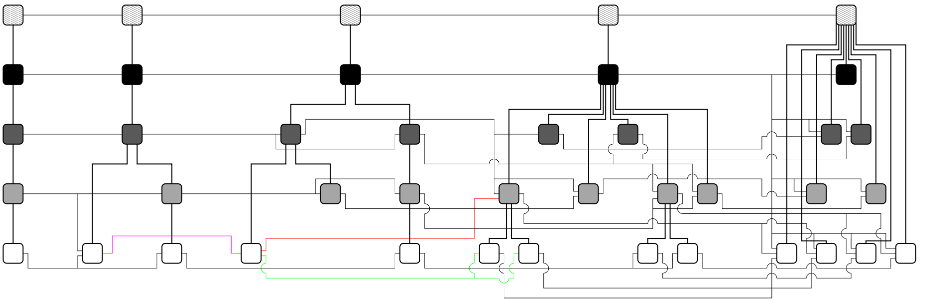


Fig. 2. Component-hypertree related to Figure 1. The set of nodes $\mathcal{H}$ is depicted by the square boxes (their colour represent the threshold value at which they appear: from black (0) to white (3), and dashed for $-\infty$). The thick lines represent the edges of $L_\downarrow$, while the thin lines represent the edges of $L_\rightarrow$. From left to right, we can observe the five Ω_i-component-trees for i from 2 to -2.

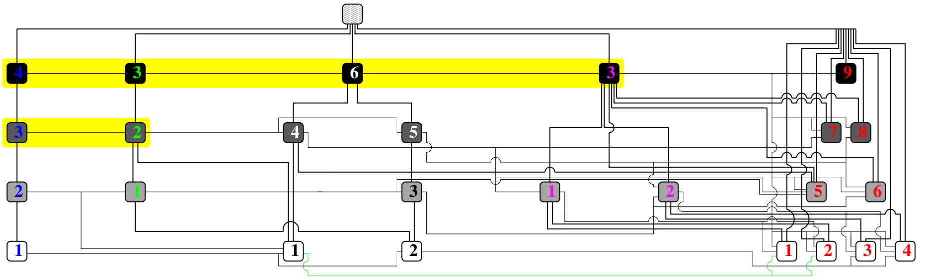


Fig. 3. Simplified component-hypertree (obtained from Figure 2) related to Figure 1. The numbers in nodes refer to the associated $\Omega_\star$-connected components in Figure 1.

processing these component-trees independently, we can compute $\mathcal{F}^\alpha$ in a global fashion in the component-hypertree (in particular, to avoid redundant work).

Based on this purpose, let us first state some remarks enabling to simplify (*i.e.*, factorise) a part of the component-hypertree data structure.

Remark 9. *The root $\mathbb{Z}^n$, which appears $u - t + 1$ times in $\mathcal{H}$ (which role is mainly to guarantee that each $\Omega_\star$-component-tree is actually a tree) only needs to be represented once*[4].

Remark 10. *Let $N_i \in \mathcal{K}_i$, $N_{i-1} \in \mathcal{K}_{i-1}$ be the same subset of $\mathbb{Z}^n$, and the subtrees of the Ω_i- and Ω_{i-1}-component-trees of root N_i and N_{i-1}, be identical. From Definition 2, we have $\mathcal{F}^\alpha(N_i) = \mathcal{F}^\alpha(N_{i-1})$. Then, several successive $\Omega_\star$-component-trees can share their similar "bottom" parts, thus reducing both space and time complexity. In particular, this is true for the $|E|$ singleton sets $\{x\}$ ($x \in E$) which need only to be represented once (for instance in the Ω_t-component-tree).*

By contrast, a node which appears in two successive $\Omega_\star$-component-trees must be represented twice if its subtrees differ (which may modify their value of c^α, and thus $\mathcal{F}^\alpha$, see yellow parts of Figure 3 for examples of such nodes).

Remark 11. *As stated in Remark 8, a node may have several decompositions. However, from an algorithmic point of view, only one decomposition is necessary. We should, in particular, preserve the decomposition which corresponds to the finer subdivision of the node,* i.e., *the one composed by the nodes at the threshold value v^+ defined in Remark 8 (see the green edge in Figure 3, for an example).*

From these remarks, a component-hypertree can be simplified to contain only nodes which are computationally useful (see Figures 2 and 3).

Note that a simplified component-hypertree preserves its principal two specificities: (i) the subgraph of $(\mathcal{H}, L_\downarrow)$ induced by the subset of nodes $\mathcal{K}_\star$ is the $\Omega_\star$-component-tree, and (ii) the subgraph of $(\mathcal{H}, L_\rightarrow)$ induced by the subset of nodes $\mathcal{S}_v$ corresponding to the $\Omega_\star$-connected components obtained at the threshold value v provides a (deterministic) hierarchical decomposition of $X_v(I)$ into Ω_i-connected components (for decreasing values of i).

4.4 Segmentation

The main purpose is now to compute $\mathcal{F}^\alpha(\mathbb{Z}^n)$ for all the $\Omega_\star$-component-trees of the component-hypertree. (Note that we restrict, in the sequel, our study to $\alpha \in \,]0, 1[$, since the cases $\alpha = 0$ or 1 can be treated in a more simple way.)

For any $N \in \mathcal{H}$, we define the following notions. If $\mathcal{F}^\alpha(N) = \{N\}$ (resp. $\mathcal{F}^\alpha(N) \neq \{N\}$), we say that N is *on* (resp. *off*). Let $d(N)$ be the set of nodes D such that $(N, D) \in L_\rightarrow$, *i.e.*, the nodes forming the coarsest partition of N in

[4] From an algorithmic point of view, it needs not to be represented since we never have $\mathcal{F}^\alpha(\mathbb{Z}^n) = \{\mathbb{Z}^n\}$, except in the useless case $\alpha = 0$ and $\prec$ is $\leq$ (when $\alpha = 0$, the case $\prec$ is $<$ should be considered, to get access to the smallest subset including M).

$\mathcal{H}$. Let $d_\bullet(N) = \{D \in d(N) \mid \mathcal{F}^\alpha(D) = \{D\}\}$ and $d_\circ(N) = d(N) \setminus d_\bullet(N)$ be the subsets of $d(N)$ composed of the nodes which are on and off, respectively. We set

$$\begin{cases} \delta^+(N) = \alpha.n(N,M) - (1-\alpha).p^*(N,M) - \sum_{N' \in ch(N)} c^\alpha(N') \\ \delta^-(N) = c^\alpha(N) - \sum_{D \in d(N)} c^\alpha(D) \end{cases}$$

Property 12. *If $d(N) = \emptyset$ (*i.e.*, $N = \{x\}$ ($x \in E$) is a singleton node), then we have $\mathcal{F}^\alpha(N) = \{N\}$ if $x \in M$ and $\emptyset$ if $x \notin M$ (see Definition 2). Moreover, we straightforwardly have $\delta^-(N) = 0$ while $\delta^+(N) = \alpha$ if $x \notin M$ and $\alpha - 1$ if $x \in M$.*

Property 13. *In the case where $d(N) \neq \emptyset$, it derives from the above definitions that Equation (2) is equivalent to the following inequality*

$$\sum_{D \in d(N)} \delta^+(D) \prec \sum_{N' \in ch(N)} \delta^-(N') \qquad (3)$$

Property 14. *In the case where $d(N) \neq \emptyset$, we have $\mathcal{F}^\alpha(N) = \{N\}$ (*i.e.*, N is on) if Equation (3) is true and $\mathcal{F}^\alpha(N) = \bigcup_{N' \in ch(N)} \mathcal{F}^\alpha(N')$ (*i.e.*, N is off) if it is false. It can also be proved that we have $\delta^-(N) = \sum_{D \in d_\circ(N)} \delta^+(N)$ if N is on, and $\sum_{N' \in ch(N)} \delta^-(N) - \sum_{D \in d_\bullet(N)} \delta^+(N)$ if N is off. Finally, we also have $\delta^+(N) = \sum_{D \in d(N)} \delta^+(D) - \sum_{N' \in ch(N)} \delta^-(N')$.*

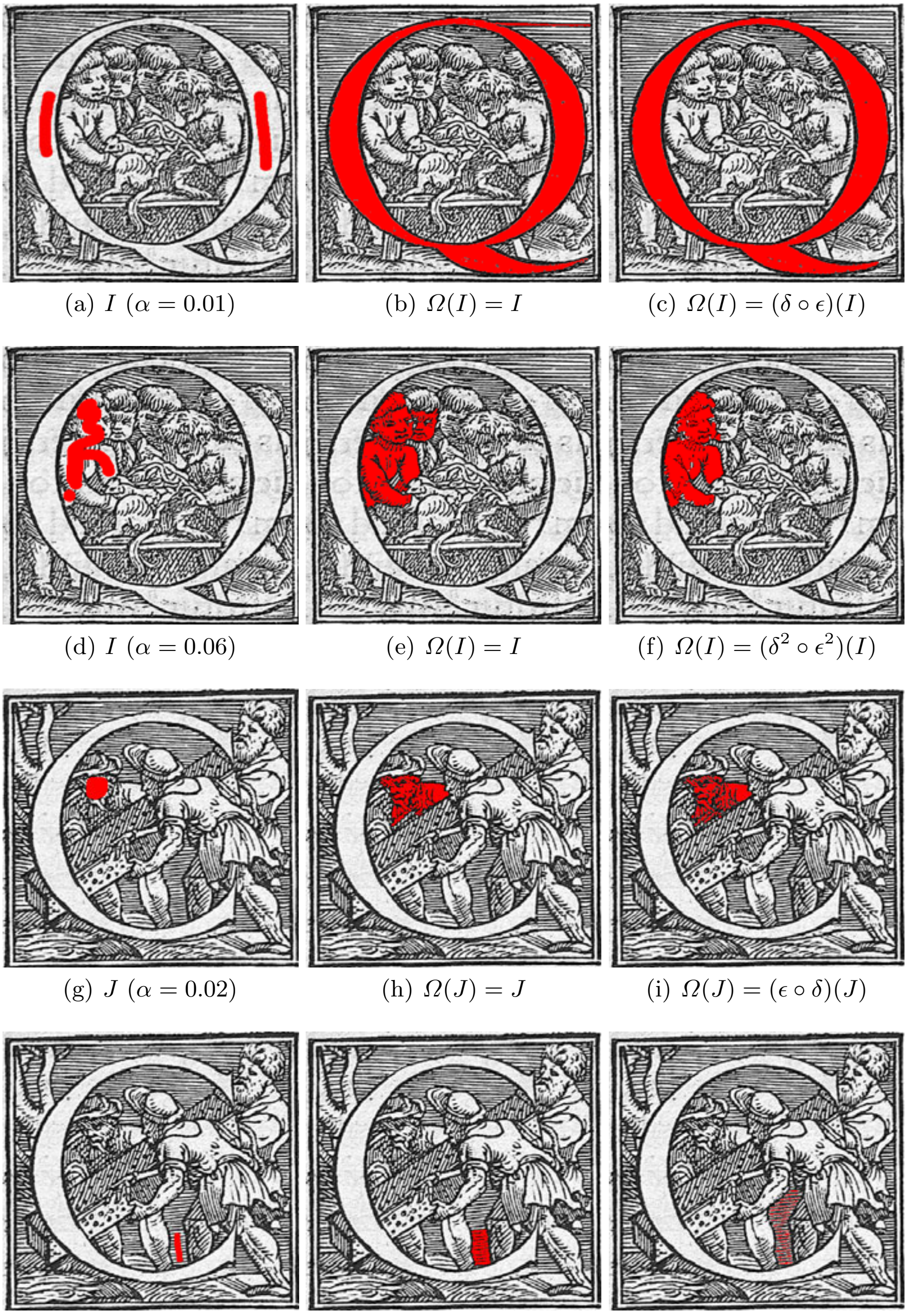
From Properties 12–14, we obtain an algorithmic process to recursively compute (in a "bottom-up/right-to-left" fashion) the sets $\mathcal{F}^\alpha(\mathbb{Z}^n)$ in the component-hypertree, by storing the values δ^-, δ^+ and performing (at most) one comparison and a few additions at each node.

4.5 Optimisation

As stated in Section 4.3, the simplification of the component-hypertree data structure provides a first way to decrease the cost of the computation of $\mathcal{F}^\alpha$. Some supplementary optimisations derive from the following properties. (These optimisations require a longer discussion which will be proposed in a further issue.)

Property 15. *When $\sum_{D \in d(N)} \delta^+(D) \prec 0$ (which happens* a fortiori*, but not necessarily, when all the nodes of d(N) are on), we have $\mathcal{F}^\alpha(N) = \{N\}$. Then, we can avoid to compute $\mathcal{F}^\alpha$, δ^- and δ^+ for the nodes $N' \in ch(N)$ and their respective subtrees. However it may (sometimes) be necessary to compute latter δ^- and δ^+ for some of these unprocessed nodes, due to the (possible) non-increasingness of M^α with respect to the increasingness of the mask images.*

When the segmentations have been computed for a given α, the computation of new segmentations, for another α, may require to process, once again (potentially) all the nodes of the component-hypertree. However, we have the following property.



(a) I ($\alpha = 0.01$)

(b) $\Omega(I) = I$

(c) $\Omega(I) = (\delta \circ \epsilon)(I)$

(d) I ($\alpha = 0.06$)

(e) $\Omega(I) = I$

(f) $\Omega(I) = (\delta^2 \circ \epsilon^2)(I)$

(g) J ($\alpha = 0.02$)

(h) $\Omega(J) = J$

(i) $\Omega(J) = (\epsilon \circ \delta)(J)$

(j) J ($\alpha = 0.03$)

(k) $\Omega(J) = J$

(l) $\Omega(J) = (\epsilon^2 \circ \delta^2)(J)$

Fig. 4. (a,d) An image I. (b,c) Results with anti-extensive masks for $\alpha = 0.01$. (e,f) Results with anti-extensive masks for $\alpha = 0.06$. (g,j) An image J. (h,i) Results with extensive masks for $\alpha = 0.02$. (k,l) Results with extensive masks for $\alpha = 0.03$.

Property 16 ([12]). *Let $0 \leq \alpha_1 < \alpha_2 \leq 1$. Then we have $M^{\alpha_2} \subseteq M^{\alpha_1}$.*

Its main consequence is the existence, for each node N, of a critical value α_c such that N is on iff $\alpha \prec \alpha_c$. Consequently, when a node N is switched on (resp. off) during the processing of the component-hypertree, at a given value α, we know that α_c belongs to $[\alpha, 1]$ (resp. $[0, \alpha]$). At each new process, the bounding $[\alpha_c^-, \alpha_c^+] \subseteq [0, 1]$ of α_c at each node can then be refined. In particular, this may enable to avoid computation at several nodes, by only checking (in the favourable cases) whether $\alpha \prec \alpha_c^-$ or $\alpha \not\prec \alpha_c^+$.

5 Experiments

The method has been applied for segmentation of drop caps from ancient documents (provided by the *Centre d'Études Supérieures de la Renaissance*[5]). These images are often noisy, which may lead to undesired connections, or disconnections when considering a standard connectivity.

In such cases, a "connectivity thresholding", may lead to improved results, thus justifying a multiscale connectivity approach. In particular, when erroneous connections appear with a standard connectivity (Figure 4(b,e)), disconnections can be obtained with anti-extensive filters (Figure 4(c,f)). Conversely, when structures of interest are disconnected with a standard connectivity (Figure 4(h,k)), reconnections can be obtained with extensive filters (Figure 4(i,l)). (Note that the filtered images $\Omega(.)$ have been computed with erosions and dilations involving a 3×3 cross structuring element.)

These examples are representative of the artifacts which may generally appear in grey-level images, and the induced consequences when they are processed by connected filtering techniques. The behaviour of the hypertree segmentation method, qualitatively assessed here, can then be considered as relevant with respect to the expected results.

6 Conclusion

A new data structure, the component-hypertree, has been presented. It models, in a compact fashion, several component-trees of a same image, induced by second-order mask-based connectivity (or, possibly, by any other kind of connectivity presenting similar properties). It may be seen, in particular, as a contribution to the concept of multiscale connectivity.

An enriched version of a segmentation method relying on component-trees has been proposed for this structure. It enables in particular to efficiently compute the segmentation of an image for several connectivities. This approach, whose relevance has been partially assessed here, by lack of space, will be described and validated in a more detailed fashion in a further issue.

It will also be considered to improve this segmentation approach by authorising the computation of results related to distinct (user-defined) connectivity levels, depending on the position in the image.

[5] `http://cesr.univ-tours.fr`

References

1. Berger, C., Géraud, T., Levillain, R., Widynski, N., Baillard, A., Bertin, E.: Effective component tree computation with application to pattern recognition in astronomical imaging. In: ICIP, pp. 41–44 (2007)
2. Braga-Neto, U., Goutsias, J.: A multiscale approach to connectivity. Computer Vision and Image Understanding 89(1), 70–107 (2003)
3. Braga-Neto, U., Goutsias, J.: Object-based image analysis using multiscale connectivity. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6), 892–907 (2005)
4. Caldairou, B., Naegel, B., Passat, N.: Segmentation of complex images based on component-trees: Methodological tools. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 171–180. Springer, Heidelberg (2009)
5. Jones, R.: Connected filtering and segmentation using component trees. Computer Vision and Image Understanding 75(3), 215–228 (1999)
6. Kiwanuka, F.N., Wilkinson, M.H.F.: Automatic attribute threshold selection for blood vessel enhancement. In: ICPR, pp. 2314–2317 (2010)
7. Kong, T.Y., Rosenfeld, A.: Digital topology: Introduction and survey. Computer Vision, Graphics, and Image Processing 48(3), 357–393 (1989)
8. Naegel, B., Passat, N., Boch, N., Kocher, M.: Segmentation using vector-attribute filters: methodology and application to dermatological imaging. In: ISMM, vol. 1, pp. 239–250. INPE (2007)
9. Naegel, B., Wendling, L.: Combining shape descriptors and component-tree for recognition of ancient graphical drop caps. In: VISAPP, vol. 2, pp. 297–302 (2009)
10. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Transactions on Image Processing 15(11), 3531–3539 (2006)
11. Ouzounis, G.K., Wilkinson, M.H.F.: Mask-based second-generation connectivity and attribute filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6), 990–1004 (2007)
12. Passat, N., Naegel, B., Rousseau, F., Koob, M., Dietemann, J.L.: Interactive segmentation based on component-trees. Pattern Recognition (in press), doi: 10.1016/j.patcog.2011.03.025
13. Salembier, P.: Connected operators based on tree pruning strategies. In: Najman, L., Talbot, H. (eds.) Mathematical Morphology: From Theory to Applications, ch. 7, pp. 179–198. ISTE/J. Wiley & Sons (2010)
14. Serra, J.: Connectivity on complete lattices. Journal of Mathematical Imaging and Vision 9(3), 231–251 (1998)
15. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(7), 1132–1145 (2008)
16. Urbach, E.R., Boersma, N.J., Wilkinson, M.H.F.: Vector attribute filters. In: ISMM. Computational Imaging and Vision, vol. 30, pp. 95–104. Springer, Heidelberg (2005)
17. Urbach, E.R., Roerdink, J.B.T.M., Wilkinson, M.H.F.: 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), 272–285 (2007)
18. Urbach, E.R., Wilkinson, M.H.F.: Shape-only granulometries and gray-scale shape filters. In: ISMM, pp. 305–314. CSIRO Publishing (2002)
19. Wilkinson, M.H.F., Westenberg, M.A.: Shape preserving filament enhancement filtering. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, pp. 770–777. Springer, Heidelberg (2001)

Fast Streaming Algorithm for 1-D Morphological Opening and Closing on 2-D Support

Jan Bartovsky[1,2,3], Petr Dokladal[1], Eva Dokladalova[2], and Michel Bilodeau[1]

[1] Center of Mathematical Morphology (CMM), Mines ParisTech,
77305 Fontainebleau, France
{petr.dokladal,michel.bilodeau}@mines-paristech.fr

[2] Laboratoire d'Informatique Gaspard Monge, Equipe A3SI, ESIEE Paris,
University Paris-Est, 93162 Noisy le Grand Cedex, France
{j.bartovsky,e.dokladalova}@esiee.fr

[3] Faculty of Electrical Engineering, University of West Bohemia,
30614 Pilsen, Czech Republic

Abstract. This paper presents a new streaming algorithm for 1-D morphological opening and closing transformations on 2-D support. Thanks to a recursive computation technique, the algorithm processes an image in constant time irrespective of the Structuring Element (SE) size, with a minimal latency and very low memory requirements, supporting various input data types. It reads and writes data strictly sequentially in the same (horizontal) scan order for both the horizontal and vertical SE. Aforementioned properties allow an efficient implementation in embedded hardware opening a new opportunity of a parallel computation.

1 Introduction

Within several last decades, the Mathematical Morphology (MM) [8] has not only evolved into a distinguished theory, but it has also settled in useful and practical applications in a general image analysis, industry control, etc. One branch of these applications is based on local feature detection, such as local texture orientation or granulometry [6].

The granulometry is a basic texture or random media analysis tool [5]. Traditionally, it stems from iterative sieving of image elements through successively coarser sieves obtaining a size distribution. A different approach was introduced by [10] who proposed a dedicated algorithm to compute the size distribution directly. Obviously, these results can be used as statistical descriptors in many modern applications based on machine learning techniques.

The recent development of computer algorithms focuses on decreasing the number of comparisons per pixel, which are evaluated sequentially, and therefore, have a large impact on its performance, allocating a large amount of memory. On the other hand, dedicated hardware systems have usually unlike limitations. The number of comparisons is not necessarily of such high importance because they can be evaluated in a parallel way in, e.g., Field Programmable Gate Arrays (FPGA), Complex Programable Logic Devices (CPLD), or even in Graphic Processing Unit (GPU). These systems are often more memory-limited in terms of the size and bandwidth than computers. In

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 296–305, 2011.

order to comply to this restriction, we assume the strictly sequential input and output data access as compulsory.

The paper is organized as follows: Section 2 outlines a brief overview of related morphological algorithms. Section 3 is a reminder of the basic notion of morphological transformations. Section 4 describes the main principle of the algorithm called Recursive Peak Elimination. Section 5 presents the functional C implementation and a pseudocode of the algorithm. Finally, Section 6 presents performance results of computer benchmarks and a comparison with other algorithms.

1.1 Novelty

We propose a new algorithm for morphological opening and closing with the following properties. The algorithm processes an image strictly sequentially in horizontal scan order with the minimal computation latency, which is inferred by the SE. The execution time is globally constant, independent of SE properties (size, orientation) and data precision (from 8-bit integer to 64-bit double float), the worst-case of which is upper bounded. Along with the very low memory requirements, which are much lower than the mere size of the image to process, these features make the algorithm especially interesting for time-critical and memory-limited hardware systems. Moreover, the sequential C implementation of this algorithm can also compete with other common morphological algorithms.

2 State of the Art

In this section, we present a basic overview of existing 1-D algorithms for morphological dilation/erosion that are used in computing openings and closings defined later in this paper. The first fast and still the most popular dilation algorithm is van Herk [3]. Although the computation complexity is independent of SE, it requires two passes: causal and anticausal, which makes any stream processing difficult. Later, Gil and Kimmel [2] improved van Herk's algorithm and reduced the number of comparison to only 1.5 per pixel. However, the computing latency depends on the SE size and the memory requirements are increased.

In [4], Lemire proposed a fast, stream-processing algorithm for causal linear SE. It also runs on floating-point data, has low memory requirements and zero latency. However, an intermediate storage of local maxima results in a random access to the input data. This problem is solved in Dokladal and Dokladalova [1] who proposed a new algorithm with very low memory requirements and zero latency. Recently, Morard *et al.* [7] proposed an algorithm for 1-D closing in arbitrary orientation in constant time using a stack for temporal storage of recent pixels. However, it uses a random access to the input and output data, which is inconvenient for hardware implementation.

Another approach of the direct opening computation was published by Van Droogenbroeck and Buckley [9]. They proposed an anchor based algorithm, where anchors are these points of the input image that remain unaffected by corresponding operation. A histogram is used for calculation that makes it dependent on the number of gray levels.

3 Basic Notions

Let δ_B, ε_B: $\mathbb{Z}^2 \to \mathbb{R}$ be a dilation and an erosion on grey-scale images, parameterized by a structuring element B, assumed flat (i.e., $B \subset \mathbb{Z}^2$), contiguous and translation-invariant, defined as

$$\delta_B(f) = \bigvee_{b \in B} f_b \; ; \qquad \varepsilon_B(f) = \bigwedge_{b \in \widehat{B}} f_b \tag{1}$$

The hat $\widehat{}$ denotes the transposition of the structuring element, equal to the set reflection $\widehat{B} = \{x \mid -x \in B\}$, and f_b denotes the translation of the function f by some scalar b. The SE B is equipped with an origin $x \in B$.

Let φ_B, γ_B: $\mathbb{Z}^2 \to \mathbb{R}$ be a closing and an opening on grey-scale images, parameterized by a structuring element B, defined as

$$\varphi_B(f) = \varepsilon_B[\delta_B(f)] \; ; \qquad \gamma_B(f) = \delta_B[\varepsilon_B(f)] \tag{2}$$

The closing and opening are dual transformations. Hereafter, our explanation is focused on the opening with no deliberate reason. Notice that the same principles can be applied to the closing in accordance to the duality property.

4 Principle of Algorithm

We describe the main principles of our algorithm in this section. In the beginning, we observe the influence of the opening transformation to a 1-D signal so that we can easily clarify the algorithm behavior afterwards.

At first, let us observe a behavior of the opening transformation (2) to a simple 1-D signal in Fig. 1. The opening literally cut off the peaks narrower than the length of the SE, hereafter referred to as L (the closing on the other hand fills the valleys narrower than L). Remark here that γ_B is invariant to the translation of the SE B so γ_B is not affected by the origin of B.

The proposed algorithm executes the peak cutting recursively, from the top downwards. Contrarily to the traditional composition (2), it handles image borders correctly so the peak of the whole size of L is eliminated, see Fig. 1.

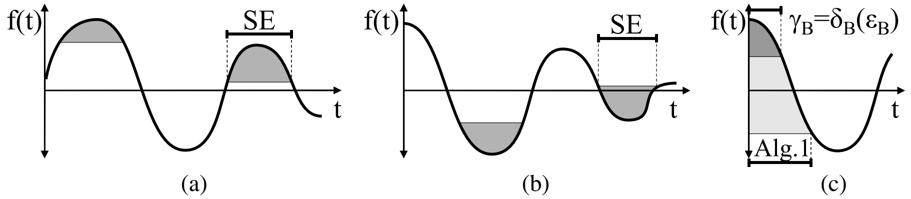


Fig. 1. Effects of opening and closing transformations on a 1-D signal: (a) opening cuts the peaks off; (b) closing fills the valleys; (c) opening of an edge: our algorithm opens the signal by the full length of the SE, L, compared to the conventional solution (2) that uses only the half of the SE length.

4.1 Recursive Peak Elimination

The Recursive peak elimination (RPE) is a part of the algorithm that manages the successive removal of signal peaks. It is composed of two tasks. At first, the input signal is continuously scanned in order to identify peaks. When a peak is encountered, the second task of peak elimination is invoked. Remark that a point of an input signal $f(x)$ is a peak if both its very precedent and subsequent points are smaller, such as

$$f(x) > f(x-1) \text{ and } f(x) > f(x+1). \tag{3}$$

The algorithm recognizes 4 basic configurations, see Fig. 2, each of them affects the algorithm behavior in a different way. Notice that Van Droogenbroeck [9] opening algorithm has 6 configurations. All configurations can be divided into 2 groups; those that characterize a peak ((a) and (b)), and those that do not ((c) and (d)).

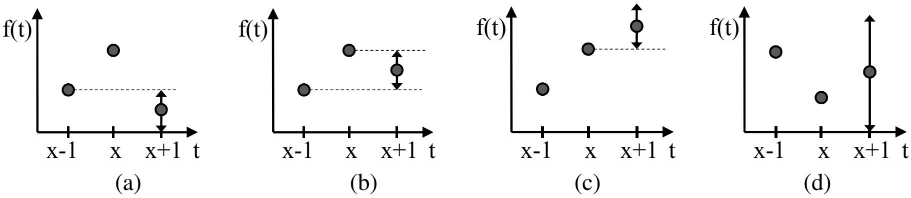


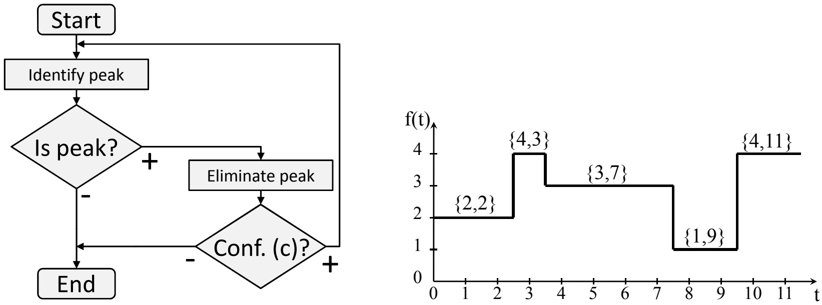
Fig. 2. Four different pixel configuration for peak identification. Conf. (a) and (b) characterize a peak, conf. (c) and (d) do not.

The peak identification process proceeds as follows. At first, $f(x)$ is compared with its precedent value $f(x-1)$. If $f(x) < f(x-1)$ (configuration (d)), the $f(x)$ value is not a peak. If not, the following value $f(x+1)$ is taken into comparison in the next step. Then if $f(x+1) < f(x-1)$ the $f(x)$ value conforms to peak configuration (a); else if $f(x+1) < f(x)$ and $f(x+1) > f(x-1)$, the peak configuration (b) is encountered. Otherwise ($f(x+1) > f(x)$), the point $f(x)$ is recognized as configuration (c), monotonously increasing function.

When a signal peak is identified, it is supposed to be removed by the peak elimination task. Obviously, the value of a signal peak has no impact of the output signal, and therefore, it can be immediately discarded out of the computation. The retrieved peak value is then replaced by one of the two neighboring values in dependence on the peak configuration. The $f(x)$ is replaced by $f(x-1)$ in configuration (a), or by $f(x+1)$ in configuration (b), respectively.

The flowchart of the whole RPE procedure is depicted in Fig. 3 a. The first step is always the identification of a peak. In the case a peak is recognized, the elimination task removes it. After the peak removal, the whole process will be repeated if, and only if, the peak is of configuration (a). The reason is that a peak may span over more than one pixel, and therefore, previous pixels must be tested on peak conditions as long as configuration (a) is recognized. The RPE iterates over previous L pixels only.

The aforementioned method needs the input data to contain a zero value duplicity. It means that two consecutive data samples must not have the same value (i.e., $f(x) \neq$



(a) Flowchart (b) Data coding

Fig. 3. (a) Flowchart of Recursive Peak Elimination. (b) Illustration of efficient data coding. Each data plateau is represented by a single pair of $\{$*value*, *last position of plateau*$\}$

$f(x+1)$). In order to handle this condition, the input data are coded into pairs of $\{$*value*, *last position of plateau*$\}$ as it is depicted in Fig. 3 b. Hence, if $f(x) = f(x+1)$ in the input data, the previously stored value of the plateau is discarded and replaced by the later one. In this case, the RPE process is not necessary, and therefore, it is not issued for current pixel $f(x+1)$. Notice that if a plateau is a peak, it will be removed in a single operation. The position also defines whether a pixel belongs to the SE that moves over data.

5 Implementation

In this section we describe the C implementation of the proposed algorithm capable of operating with horizontally- and vertically-oriented 1-D SE. In both cases, the input and output data access is strictly sequential in the horizontal scan order (each line is read from the left to the right, line by line). For the sake of simplicity, let us consider the horizontal SE first.

The implementation uses the double-end FIFO (First In First Out) structure called queue that serves as an elementary memory block to store the previously received values. It allows us to read and delete values from both ends of an ordinary FIFO.

The algorithm is composed of 2 parts. The first is an executive part that contains all the necessary commands to process one input data sample and output one result data sample according to the before mentioned principles. It manages the data-flow, borders handling, and the whole computation itself. See Alg. 1 for a pseudocode listing. The second one is an iterative, double for-loop part that is intended to call Alg. 1 for each point of the input image the listing of which is obvious, and therefore, omitted. For correct handling of borders, the input image area is extended by padding of size L in the direction of the SE orientation. For example, considering SVGA image 800 $\times$ 600 px, horizontal SE 25 px, the image area for Alg. 1 is 825×600.

The single call of Alg. 1 proceeds as follows. At first, the large while-loop removes the input data value duplicity, and executes the RPE. As it was described at section 4.1, it comprises of the peak identification using 4 configurations and the peak elimination. The while condition $F \leq$ Q.back()[1] terminates the RPE when a non-peak

Algorithm 1. $Y \leftarrow$ 1D_OPEN $(F, rp, SE,$ Q)

```
Input: $F$ - input signal sample $f(rp)$; $rp$ - actual reading position; $L$ - SE size; Q -
       pointer to actual Queue
Result: $Y$ - sample of opening $\gamma[f(t)](rp - L)$
Data: Q - a Double-End Queue
      Q.back()[] - access the last enqueued pair $\{F, rp\}$
      Q.back(2)[] - access the second to the last enqueued pair $\{F, rp\}$

1  while $F \leq$ Q.back()[1] do
2      if $F =$ Q.back()[1] then
3          Q.dequeue() ;                     // Dequeue constant value, prevent value duplicity
4          break ;                                                       // Exit while loop
5      else
6          if Q.back(2)[1]<Q.back()[1] then
7              if $F <$ Q.back(2)[1] then
8                  Q.back(2)[2] ← Q.back()[2] ;            // Copy peak position, conf. (a)
9              Q.dequeue() ;                               // Peak elimination, conf. (b), (a)
10         else
11             break ;                                          // Exit while loop, conf. (d)
12 Q.push($\{F, rp\}$) ;                                        // Enqueue current sample
13 if $rp =$ Q.front()[2] + $L$ then
14     Q.pop() ;                                                 // Delete outdated value
15 if $rp \geq L$ then
16     return (Q.front()[1] ) ;                                     // Return valid value
17 else
18     return ({}) ;                                                     // Return empty
```

configuration (c) (or possibly (d)) is encountered. The condition on line 2 retrieves the last stored value from the queue if it is equal to the current sample to prevent the value duplicity. The following condition on line 5 evaluates two previously stored pixels to exclude configuration (d). If this condition is fulfilled, the last stored pixel is identified as a peak. The peak is then retrieved by Q.dequeue() on line 9. The small exception is configuration (a) that needs the peak position Q.back()[2] to rewrite the position of the pixel Q.back(2)[2] whose value replaces the peak, see line 8.

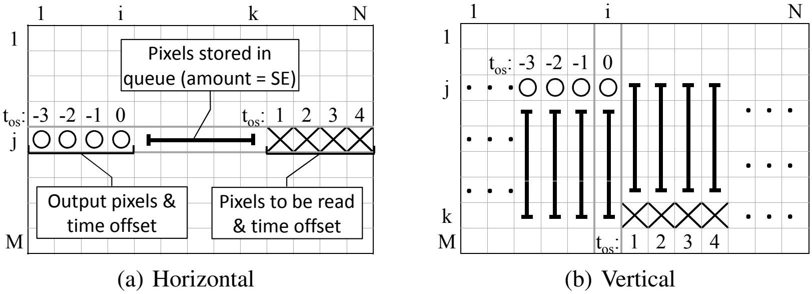
After a potential peak is eliminated, the current pixel value is unconditionally pushed into the queue along with actual reading position on line 12. The oldest stored pixel is checked whether it has been stored in the queue for too long. This check is carried out by comparing the stored reading position plus L with actual rp on line 13. Outdated values are immediately deleted. The result of Alg. 1 is always the oldest stored value if rp already exceeds L (code line 15).

5.1 Horizontal and Vertical Orientation

The algorithm described so far considers only horizontal SE due to the horizontal scan fetches the input data in the same order as the algorithm needs. In order to use the vertical SE in the same way, all existing algorithms either use a vertical data scan or transpose the image. Both result in increased memory usage and thus reduced performance. We, on the other hand, modified the algorithm to work with a vertical SE in a horizontal scan order instead.

With horizontal orientation, the algorithm reads the scan-ordered input image, processes it according to Alg. 1, and outputs the scan-ordered results as soon as they are available. An example of the image situation while processing the result of a transformation at the position $[i,j]$ is shown in Fig. 4. The computation latency is defined as a distance in the image between the actual input and output value, and it is kept at the minimal value of L (see distance of columns $k - i$ in Fig. 4).

On the other hand, the combination of the horizontal scan and the vertically oriented SE disallows the computation in a single queue. Because the horizontal scan approaches columns in a cyclic pattern (1, 2, .. N, 1 etc.), each column requires an independent queue where the computation from the corresponding column is carried out separated of each another. The executive Alg. 1 remains unchanged; the pointer to queue is its input parameter. Fig. 4 depicts an image while outputting the result at the position $[i,j]$. Note that the latency is again defined by L (lines $k-j$), but it is further multiplied by the width of the image N. Nevertheless, it is the minimal achievable latency considering the orientations of the SE and the scan order.


1
i
k
N
1
j
M
Pixels stored in queue (amount = SE)
t_{os}: -3 -2 -1 0
t_{os}: 1 2 3 4
Output pixels & time offset
Pixels to be read & time offset
1
i
N
1
j
k
M
t_{os}: -3 -2 -1 0
t_{os}: 1 2 3 4

(a) Horizontal (b) Vertical

Fig. 4. Input image when processing a pixel at location $[i,j]$ with (a) a horizontal SE, and (b) a vertical SE. × denotes the pixels to be read in next iterations, ∘ denotes the previously output pixels.

The proposed algorithm targets the memory limitation. The only memory elements are queues whose depth and amount are implied by the SE size L and image width N. For example, let us consider an opening of 8-bit, SVGA image, i.e., 800×600 px by a SE of 41 px. The queues occupy only 588 bits for horizontal, and 469 kbits for vertical SE, respectively, compared to the mere size of the image 3.66 Mbits which is never stored.

6 Experimental Results

We present the processing execution time benchmarks to illustrate the computational complexity in this section. We used the g++ compiler with -O3 optimization flag. The measurements were performed on Intel Xeon E5620 @2.4GHz CPU, with 24 GB RAM at 1333 MHz, running Linux. The time reported in tables below refers to the user CPU time consumed by the respective algorithms. We use the mountain natural photo as a testing image, originally introduced at [9], see Fig. 5.

Fig. 5. Natural photo testing image: mountain.

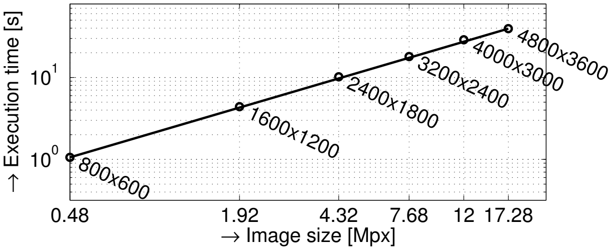


Fig. 6. Execution time of opening versus the image size. Structuring element is vertical L =101 px.

At first, we evaluated the execution time benchmark with respect to the increasing image size, see Fig. 6. The results demonstrate that the complexity of our algorithm is linear with the image size. The second benchmark in Fig. 7 retains the same image size and changes the length of the horizontal SE to show that the size of the SE L does not affect the performance, observe the constant execution time values for $L = 11$ to 101 px. However, the execution time is dependent on the image content.

The same Fig. 7 also illustrates the comparison with other dilation/erosion algorithms (naive, van Herk [3], Lemire [4], Gil and Kimmel [2], and Dokladal and Dokladalova [1]) that form an opening by composition (2), and opening algorithm (Van Droogenbroeck [9] and Morard *et al.* [7]). Our algorithm brings better results than other dilation/erosion algorithms, but worse results than opening/closing algorithms. On the other hand, neither Van Droogenbroeck nor Morard *et al.* can process streamed data due to the random access to data. Van Droogenbroeck further uses a histogram that becomes computationally very demanding with increasing number of grey levels.

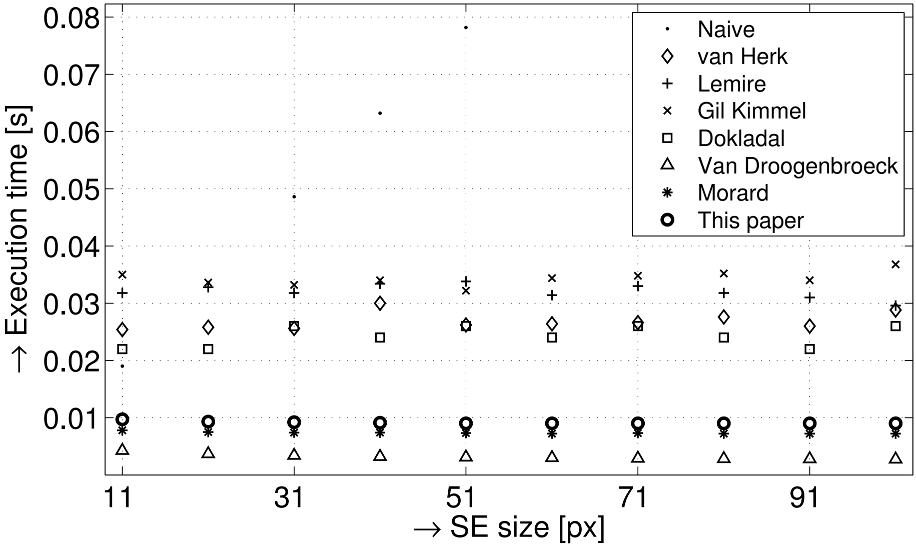


Fig. 7. Execution time of opening versus the size of the horizontal SE L. Natural photo 800 × 600 px is used.

Table 1 presents the execution time for different images. The best result is achieved with a constant image, which is not changed by opening $\{\varphi_B I = I \mid I = \text{const.}, \forall B\}$, the most demanding image is that containing random noise.

Table 1. Execution time of opening versus the image type. Structuring element is horizontal or vertical L =101 px; image size is 800 × 600 px.

Image type	constant	natural	random noise
Execution time, horizontal SE [ms]	5.1	9.0	9.3
Execution time, vertical SE [ms]	5.2	10.6	12.4

The last experiment reveals the influence of the image data precision to the execution time. The results suggest that although the performance is slightly lower for long integer and floating point data formats, the absolute difference is not significant.

Table 2. Execution time of opening versus the data type. Structuring element is vertical segment L =101 px; image size is 800 × 600 px.

Data type	char	short	int	long	float	double
Bit length	8	16	32	64	32	64
Execution time [ms]	10.7	10.5	10.6	11.6	11.3	11.9

7 Conclusions

This paper deals with a new algorithm for 1-D morphological opening and closing transformations with beneficial properties: The computational complexity is linear w.r.t. image size and independent of the SE size. The operator uses strictly sequential access to data in horizontal scan order for both vertical and horizontal SE. Along with the low latency and low memory requirements, these features make algorithm suitable for diverse real-time HW application, as well as it brings good performance in the sequential C implementation.

The future extension and undergoing work are two: 1) a rotation of the SE under arbitrary angle enables another group of applications, such as dominant angle, orientation distribution etc. 2) dedicated FPGA and GPU implementations taking advantage of the proposed algorithm that is tailored just for this purpose. Such a system shall be able to compute the whole (size, orientation) distribution during a single image scan with minimal latency.

References

1. Dokládal, P., Dokládalová, E.: Grey-scale morphology with spatially-variant rectangles in linear time. In: Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2008. LNCS, vol. 5259, pp. 674–685. Springer, Heidelberg (2008)
2. Gil, J., Kimmel, R.: Efficient dilation, erosion, opening, and closing algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 24(12), 1606–1617 (2002)
3. van Herk, M.: A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. Pattern Recogn. Lett. 13(7), 517–521 (1992)
4. Lemire, D.: Streaming maximum-minimum filter using no more than three comparisons per element. Nordic Journal of Computing 13(4), 328–339 (2006)
5. Matheron, G.: Random sets and integral geometry [by] G. Matheron. Wiley, New York (1974)
6. Menotti-Gomes, D., Najman, L., de Albuquerque Araújo, A.: 1D Component tree in linear time and space and its application to gray-level image multithresholding. In: Proceedings of 8th ISMM, vol. 1, pp. 437–448. INPE (2007), http://ismm.dpi.inpe.br/2007/
7. Morard, V., Dokladal, P., Decenciere, E.: Linear openings in arbitrary orientation in O(1) per pixel. In: ICASSP 2011: Acoustics, Speech, and Signal Processing (2011)
8. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, Inc., Orlando (1983)
9. Van Droogenbroeck, M., Buckley, M.J.: Morphological erosions and openings: Fast algorithms based on anchors. J. Math. Imaging Vis. 22(2-3), 121–142 (2005)
10. Vincent, L.: Granulometries and opening trees. Fundam. Inform. 41(1-2), 57–90 (2000)

Hierarchical Analysis of Remote Sensing Data: Morphological Attribute Profiles and Binary Partition Trees

Jon Atli Benediktsson[1], Lorenzo Bruzzone[2], Jocelyn Chanussot[3],
Mauro Dalla Mura[1,2], Philippe Salembier[4], and Silvia Valero[3,4]

[1] Faculty of Electrical and Computer Engineering,
University of Iceland, Reykjavik, Iceland
benedikt@hi.is

[2] Department of Information Engineering and Computer Science,
University of Trento, Trento, Italy
bruzzone@ing.unitn.it, dallamura@disi.unitn.it

[3] GIPSA-Lab, Grenoble Institute of Technology, France,
jocelyn.chanussot@gipsa-lab.grenoble-inp.fr,
silvia.valero-valbuena@gipsa-lab.grenoble-inp.fr

[4] Technical University of Catalonia (UPC), Barcelona, Catalonia, Spain
philippe.salembier@upc.edu

Abstract. The new generation of very high resolution sensors in airborne or satellite remote sensing open the door to countless new applications with a high societal impact. In order to bridge the gap between the potential offered by these new sensors and the needs of the end-users to actually face tomorrow's challenges, advanced image processing methods need to be designed. In this paper we discuss two of the most promising strategies aiming at a hierarchical description and analysis of remote sensing data, namely the Extended Attribute Profiles (EAP) and the Binary Partition Trees (BPT). The EAP computes for each pixel a vector of attributes providing a local multiscale representation of the information and hence leading to a fine description of the local structures of the image. Using different attributes allows to address different contexts or applications. The BPTs provide a complete hierarchical description of the image, from the pixels (the leaves) to larger regions as the merging process goes on. The pruning of the tree provides a partition of the image and can address various goals (segmentation, object extraction, classification). The EAP and BPT approaches are used in experiments and the obtained results demonstrate their importance.

1 Introduction

Satellite and airborne remote sensing is currently undergoing a technical revolution with the appearance and blooming development of very high resolution sensors, the term *resolution* having the following three meanings:

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 306–319, 2011.

- *Spatial resolution*: Metric and sub-metric resolutions are now currently available for satellite remote sensing. A high spatial resolution opens the door for very accurate geometrical analysis of objects present in scenes of study.
- *Spectral resolution*: After decades of use of multispectral remote sensing, most of the major space agencies now have new programs to launch hyperspectral sensors, recording the reflectance information of each point on the ground in hundreds of narrow and contiguous spectral bands. The spectral information is instrumental for the accurate analysis of the physical component present in one scene.
- *Temporal resolution*: Due to the launch of constellations of satellites and the increasing number of operating systems, the temporal resolution between two acquisitions over a given scene of interest has dramatically decreased. This opens the door to the accurate monitoring of abrupt changes and to efficient response in case of major disasters. Temporal phenomena with longer scales may also be monitored.

The accurate analysis of remote sensing images is an important task for many practical applications with high societal impact, such as precision agriculture, monitoring and management of the environment, urban planning, natural hazards and disasters management, security and defense issues. However, in order to fully exploit the potential offered by the new generations of sensors and to actually face all the emerging applications, advanced image processing methods are required. As a matter of fact, most of the traditional processing algorithms fail when the resolution increases significantly. For instance, statistical learning becomes intractable with hyperspectral data because of the dimensionality of the data. Similarly, while it was easy to classify urban versus non urban areas with medium resolution data, very high resolution data enable the accurate classification at the building scale, but this requires to completely redesign the whole processing chain.

While the spectral information is usually used to perform a pixel-wise classification of the data based on the physical properties of the sensed materials, extracting meaningful *spatial information*, characterizing the sensed landscape in a complementary way with respect to the spectral signatures of the land covers, is a challenging task for an accurate analysis of the structures in the image. Recently, Daya Sagar and Serra [1] underlined how the retrieval and characterization of the spatial information is a current challenge for geoscience scientists. Due to the wide range of features related to the spatial domain, there are several ways of characterizing this source of information. From a general survey of techniques modeling the spatial information in remote sensing, one can notice that there are different approaches for extracting the spatial information and correspondent ways (with different levels of abstraction) for including the extracted information in the processing chain aiming at the understanding of the image. Roughly, it is possible to group the techniques in three approaches (ordered increasingly according to the level of semantic introduced in the representation of the scene):

1. techniques aiming at modeling the spatial context at a pixel-level by looking at the neighborhood of each pixel [2, Chap. 8],[3, Chap. 6],[4,5,6,7];
2. techniques based on segmentation that exploits the partition of the image into regions by extracting spatial features that can describe the structures in the scene [8, Chap. 5][9,10,11,12,13,14,15];
3. techniques working at the object level where also the thematic characteristics and the relations between structures are taken into account [16,17,18,19,20].

It this scenario, Mathematical Morphology (MM) [21,22,23,24] holds a fundamental role since it provides a set of powerful tools for analyzing the spatial domain.

In 2002, Soille and Pesaresi [25] identified the main applications in the context of remote sensing image analysis that could be addressed by MM: i) image filtering; ii) image segmentation and iii) image measurements. Thus, MM tools permit to enrich the image analysis by including spatial information mainly at pixel- and region-level, and, in the decade following this milestone paper, numerous techniques involving MM for the analysis of remote sensing images have been proposed. In particular, focusing the attention on very high resolution (VHR) images, we highlight the consolidation of the role of *connected operators* [26,27] as efficient filters for achieving a simplification of the image obtained by only merging flat zones (i.e., avoiding the detriment of the geometrical features of the regions unaffected by the transformation). Connected operators have gained popularity also due to the successful diffusion of Morphological Profiles (MPs) [9,28]. MPs are a multiscale decomposition of a grayscale image in a stack of filtered images obtained by transforming the input scene with a sequence of opening and closing by reconstruction filters (i.e., connected operators) based on structuring elements (SEs) with fixed shape and increasing size. In [25] it was also fostered that multiscale and multidirectional segmentation methods based on the concept of MP were promising approaches since they could lead to a further exploitation of MM tools in the remote sensing field. Accordingly, not only the multiscale or multidirectional approaches have confirmed their suitability to the extraction of the spatial information but furthermore, techniques performing more general *multilevel analyses* have been proposed.

Moreover, with a further step forward on the path leading to the semantical understanding of the scene, *hierarchical representations* of structures in the image have started to be successfully exploited[29,30]. In particular, the use of Binary Partition Trees (BPT) [31] has been recently investigated in remote sensing for various applications (segmentation, classification, object detection) [32,33,34,35].

This paper is organized as follows. General considerations on the multilevel and hierarchical approaches are given in the next section. Sec. 3 is devoted to the presentation of the Attribute Profiles, a generalization of MPs based on attribute connected filters. Sec. 4 is devoted to the presentation of Binary Partition Trees and their use for the analysis of hyperspectral data. Finally, concluding remarks are presented in Sec. 5.

2 Multilevel Analyses and Hierarchical Representations of the Scene

The intrinsic mixture of land covers in natural landscapes can lead to a very complex imaged scene especially when dealing with dense urban areas and VHR images. Multiscale approaches have proved to be suitable for extracting the components relevant for the application, crawling the overwhelming information given by the huge amount of details [36,10]. The MP leads to a multiscale decomposition of the image (in bright and dark components) since it can be seen as a sequential configuration of the scene with a progressively decreasing amount of either bright or dark details [9]. Thus, by considering the behavior of the grayscale value of each pixel as the size of the SE varies, it is possible to extract information on the scale (i.e., size) of the objects in the image. Using linear SEs with different directions (i.e., morphological directional profiles [25]) enables the characterization of the structures on the basis of their orientation and length. Another recent development dealing with advanced morphological directional operators applied to remote sensing data is the use of path operators for the detection of the road network on VHR remote sensing images [37]. With the presence of heterogeneous structures in the image, a multiscale or multidirectional approach is compulsory since an analysis carried out at a single scale/direction would lead to a partial extraction of the spatial characteristic of interest. However, apart from the scale and direction, other parameters can be used for a more complete characterization of the objects in the scene (e.g., for modeling the shape or texture). Extending the multiscale idea, if the image is progressively simplified by performing a sequence of transformations with a varying parameter, it is possible to obtain a more general multilevel decomposition of the image. Attribute filters [38] proved to be suitable for implementing this idea. If applied in a sequence with fixed attribute and varying reference used in the definition of the predicate we obtain Attribute Profiles (APs) [39,40], which can be considered as a generalization of the MPs. By exploiting the use of attribute filters in the AP structure, it is possible to perform a multilevel analysis (i.e., decomposition) of the image according to many possible characteristics (e.g., geometric, textural, spectral, etc.) due to the freedom in the definition of the predicate. Moreover, according to the attribute, it is possible to extract features on the scene whose significance is closer to the conceptual information that is sought (e.g., the attribute area can better fit the general concept of size than might the width of a SE).

In the framework of mathematical morphology, representations of an image as a hierarchical tree structure of connected components have been proposed. Examples of hierarchical structures are min- and max-trees [41], inclusion-trees (or tree of shapes) [42] and binary partition trees (BPTs) [31].

Min- and max-trees are based on the region inclusion obtained by performing a threshold decomposition [43] of the image. Inclusion tree relies on the saturation operator, which basically fills the holes in the regions. BPTs store a hierarchical region-based representation in a tree structure. This provides a hierarchy of regions at different levels of resolution to cover a wide range of applications. This generic representation, independently from its construction, can be used in many different applications such as segmentation, classification, indexing, filtering, compression or object recognition [31,35]. Depending on the definitions of the used region model and the distance used to determine the order of merging of regions, different BPTs can be constructed. Once a BPT is constructed, providing a full hierarchical representation of the information, the pruning step must be defined in order to either segment simplify the image or to select one given node (object detection).

Recently, such structures have started to be exploited also in remote sensing, mainly for image classification and segmentation [39,40,44,32,33,34,35,45,46]. The hierarchical representation of the images is not only useful for computing efficient algorithmic implementations of some MM operators [47] but can give important information on the relations between the regions in the image, since the nodes in the tree can refer to salient objects in the image.

3 Attribute Profiles and Extended Attribute Profiles

In this section we review the definition of a generalization of the concept of the MP, *i.e* the Attribute Profile [40] and its extension for multichannel data, the Extended Attribute Profile [44]. For a review of ten years of developments of the MP, the reader is referred to [48].

3.1 Attribute Profiles

Attribute profiles were proposed in [40] for overcoming the limitation of the MP to model other feature than the size of the objects. APs are based on attribute filters [38] and thus, can process the image according to features such as the contrast, texture, geometry, etc. Analogously to the MP, the AP operates either on bright or dark component with attribute thinning and attribute thickening as operators, respectively. The AP can be defined as a concatenation of a thickening attribute profile, $\Pi_{\phi T'}$, and an thinning attribute profile, $\Pi_{\gamma T'}$ computed with a generic ordered criterion T':

$$AP(f) = \left\{ \begin{array}{lll} \Pi_{\phi^{T'_\lambda}}, & \lambda = (n-1+i), & \forall \lambda \in [1, n]; \\ \Pi_{\gamma^{T'_\lambda}}, & \lambda = (i-n-1), & \forall \lambda \in [n+1, 2n+1]. \end{array} \right\}. \tag{1}$$

With $T' = \{T_1, T_2, \ldots, T_n\}$ the set of ordered criteria, for $T_i, T_j \in T'$ and $j \geq i$, the relation $T_i \subseteq T_j$ holds. The family of criteria needs to be ordered for guaranteeing that the absorption property is fulfilled by the AP (condition that might not be verified for non increasing predicates). The fulfillment of the absorption property ensures the consistency of the derivative of the AP (DAP). Different information can be extracted from the structures in the scene according to the attribute and criterion considered in the filtering leading to different multilevel decompositions of the image. Moreover, the computation of the APs, when based on the min- and max-tree representation of the image, leads to an efficient implementation of the multilevel filtering. In particular, an AP can be obtained by building up only once a max- and a min-tree for the thinning and thickening transformations, respectively, and by performing each filtering of the sequence as a different pruning of the tree.

3.2 Extended Attribute Profiles (EAP)

In [44] the AP was extended to multichannel images as proposed in [49]. Thus the EAP is obtained by computing an AP on each of the c principal components extracted from the original multichannel data (e.g., hyperspectral image):

$$EAP(f) = \{AP(PC_1), AP(PC_2), \ldots, AP(PC_c)\}. \tag{2}$$

When considering different attributes, it is possible to stack in the same data structure the EAP computed with each attribute, leading to the definition of Extended Multi-Attribute Profile (EMAP) [44]:

$$EMAP(f) = \{EAP_{a_1}(f), EAP'_{a_2}(f), \ldots, EAP'_{a_m}(f)\} \tag{3}$$

with a_i a generic attribute and $EAP' = EAP \backslash \{PC_1, \ldots, PC_c\}$ for avoiding the multiple presence of the c principal components.

APs and EAPs were used for the thematic classification of panchromatic VHR images [40] and hyperspectral images [44] proving that the extraction of different spatial features can lead to greater accuracies in comparison with those obtained by considering MPs and EMPs, respectively. In [50], the APs were considered for change detection on VHR images showing promising results in providing a characterization of the changes complementary to the one given by the classical spectral analysis.

4 Binary Partition Trees for the Analysis of Hyperspectral Data

Hyperspectral sensors collect multivariate discrete images in series of narrow and contiguous wavelength bands. The resulting data sets enable the characterization of regions based on their spectral properties. Conventional analysis techniques have traditionally considered these images as an unordered array of spectral measurements. In the last few years, the importance of the spatial information considering, in particular, spatial correlation and connectivity in the image has been proved. As previously mentioned, this information turns out to be essential to interpret objects in natural scenes. Hence, hyperspectral analysis tools should take into account both the spatial and spectral spaces. However, the number of wavelengths per spectrum and pixel per image as well as the complexity of handling spatial and spectral correlation explain why this approach is still a largely open research issue.

Over the past decade, Binary Partition Trees have been used for various purposes in various contexts for grey scale and color images. Due to the high dimensionality of the data, extending the use of BPTs to hyperspectral images is a very challenging issue. This has been recently investigated, as a way to provide an abstraction from the pixel-spectrum-based representation[32]. Note that the use of BPTs has also recently been investigated in the frame of polarimetric SAR images filtering and segmentation [51]. This representation [31] hierarchically stores a region-based representation in a tree structure, as illustrated on Fig. 1, and provides a hierarchy of regions at different levels of resolution to cover a wide range of applications. This generic representation can be based on an iterative region merging algorithm but requires a region similarity metric and a region model. The *region model* M_R specifies how regions are represented and how to model the union of two regions. The *merging criterion* $O(R_i, R_j)$ defines the similarity between neighboring regions and hence determines the order in which regions are merged. Working with hyperspectral data, the definition of both concepts is not straightforward. Regarding the region model, a non parametric statistical model (a multi-dimensional histogram) is used [52]. This leads to the definition of a robust distance between histograms taking into account the correlation between bands. Different hyperspectral region models and similarity metrics are presented and analyzed in [33] and a new merging strategy using a new association measure depending on canonical correlations relating principal coordinates is proposed in [35].

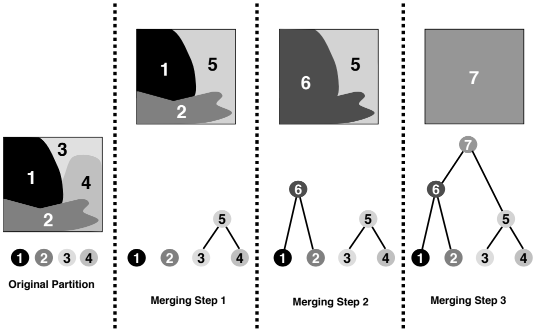


Fig. 1. Example of BPT construction using a region merging algorithm

Once the BPT is constructed, a second step consists in designing a pruning strategy meeting the goal of the addressed application. These two steps (construction and pruning of the BPT) are separate: the construction is based on the intrinsic information of the image, while the pruning should be related to the application. A new pruning strategy aiming at the segmentation of hyperspectral images is proposed in [34]: the regions contained in the BPT branches are studied by recursive spectral graph partitioning. The goal is to remove subtrees composed of nodes which are considered to be similar. To this end, affinity matrices on the tree branches are computed using a new distance-based measure depending on canonical correlations relating principal coordinates.

4.1 Experimental Analysis

4.2 Experiments Based on APs and EAPs

In [40] the APs were used for extracting spatial features considered for the classification of two Quickbird panchromatic images acquired on the city of Trento (Italy). The APs were computed with three attributes: i) area; ii) moment of inertia [53]; and iii) standard deviation. The area attribute was chosen for modeling the size of the structures in the image, the moment of inertia for extracting information on the shape of the regions and the standard deviation was considered as a descriptor of the spectral homogeneity of the objects. In the experiments, each AP was firstly classified by a Random Forest (RF) classifier [54] separately and then all the APs stacked together were considered. The use of different attributes led to the extraction of complementary information from the scene leading to increasing accuracies when considered in classification. In terms of classification errors, a decrease in the kappa error up to 38% and 17% with respect to the original panchromatic image and the MP, respectively was experienced when considering the APs.

The EAPs were used in [44] for the classification of two hyperspectral images acquired on Pavia. Four attributes were considered in the analysis by building the four correspondent EAPs: i) area of the regions; ii) diagonal of the box bounding the region; iii) moment of inertia; iv) standard deviation of the gray-level values of the pixels in the regions. All the EAPs computed were also considered together in the EMAP structure. A RF classifier was employed for classifying the features extracted by the profiles. The inclusion of the spatial information led to an increase in accuracy of up to 21.9% with respect to considering only the PCs (spectral information only). In the experiments the use of the proposed EAPs and EMAP led to an increase of overall accuracy up to about 12% over the results obtained by considering the EMP. Particulars of the classification maps obtained are shown in Fig. 2.

4.3 Experiments Based on BPTs

We present here some results dealing with segmentation, object detection and recognition as these are important challenges in remote sensing images. The automated selection of results in hierarchical segmentations combining spectral/spatial information has been previously studied [30]. Despite of some interesting results, problems regarding under and over-segmentation remained. Adequately pruning BPT representations combining spectral and spatial features can overcome some of these limitations.

We first provide an evaluation of the BPT pruning proposed in [34]. The experiments have been performed using a portion of Pavia Center image from hyperspectral ROSIS sensor. The data contain 102 spectral bands. Fig. 3(a) shows a RGB combination of three of them. The BPT is computed by the procedure described in [34]. To evaluate the quality of the BPT pruning, we compare the results obtained with a Min cut applied on the BPT against a trivial pruning

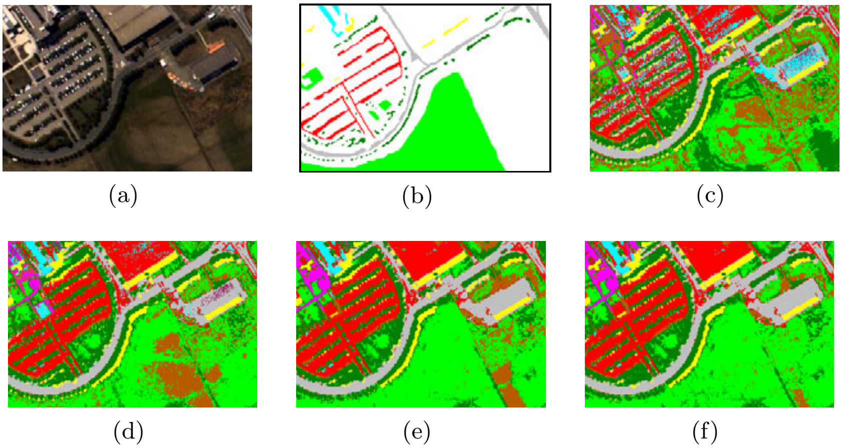
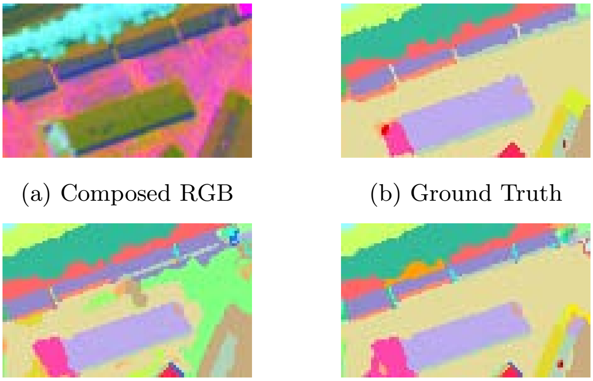


Fig. 2. Pavia data set. (a) True color composition; (b) Reference map. Details of the classification maps obtained with a RF classifier and: (c) the PCs, (d) the EMP, (e) the EAP with area attribute, and (f) the EMAP.
Thematic classes: 🟩 trees, ⬜ asphalt, 🟪 bitumen, 🟦 gravel, 🟪 metal sheets, 🟨 shadows, 🟩 meadows, 🟥 self-blocking bricks, 🟫 bare soil.

criterion based on the number of regions in the BPT following the merging sequence [33]. To evaluate the resulting partitions, the symmetric distance d_{sym} [55] is computed with the manually set ground truth (GT) shown in Fig. 3(b). Fig. 3(c)(d) show the segmentation results obtained with the trivial and the Mincut BPT pruning, respectively. In both cases, the partitions have 54 regions. Comparing both results, the quantitative d_{sym} and the visual evaluation corroborate that the partition obtained by the advanced pruning is much closer to the ground truth than the one computed with a simple stopping of the region merging algorithm.

The second set of experiments is performed using a portion of a publicly available HYDICE hyperspectral image. After removing water absorption and noisy bands, the data contain 167 spectral bands. Fig. 4(a) shows a RGB combination of three of them. To evaluate the quality of the BPT construction (and not the pruning strategy), we extract a segmentation result involving a given number N_R of regions by undoing the last $N_R - 1$ mergings over the initial partition. The result is compared with the classical Recursive Hierarchical Segmentation algorithm (RHSEG), the similarity criterion used for RHSEG being SAM with spectral clustering weight 0.1 [29,11]). The manually created GT is shown in Fig. 4(b). Fig. 4(c)(d) show the segmentation results obtained with BPT and RHSEG, respectively. In both cases, the resulting partitions involve 63 regions. Again, the qualitative visual inspection and the quantitative evaluation assess the interest of the BPT representation.

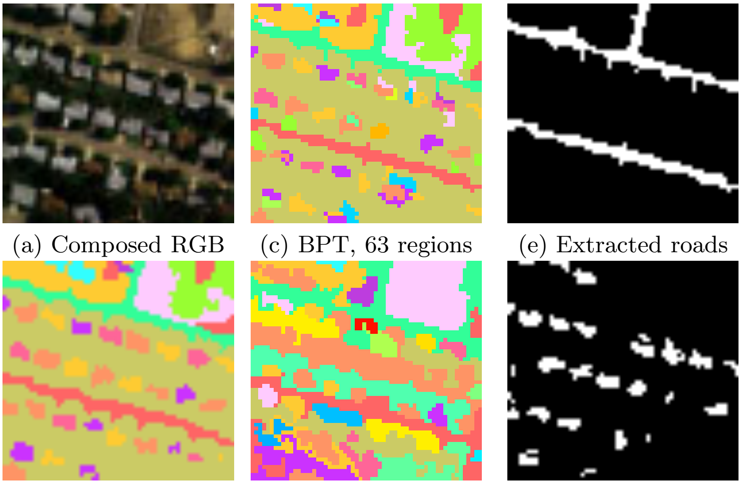
Finally, object detection and recognition is also considered. By combining simple shape descriptors (area, elongation of an oriented bounding box) with spectral information (typical spectrum for one given class), one can select



(a) Composed RGB (b) Ground Truth

(c) Pruning by number of regions (d) Min *Ncut* Pruning BPT

Fig. 3. (a) Pavia Center ROSIS RGB Composition, (b) Manually created Ground Truth, (c) Partition extracted from the trivial pruning leading to d_{sym}=40, (d) Partition computed with the proposed pruning leading to d_{sym}=20



(a) Composed RGB (c) BPT, 63 regions (e) Extracted roads

(b) Ground Truth (d) RHSEG, 63 regions (f) Extracted buildings

Fig. 4. a) Urban HYDICE RGB Composition, (b) Manually created Ground Truth, (c) Partition extracted from BPT leading to d_{sym} =25, (d) Partition computed with RHSEG [29] leading to d_{sym} =70, Roads (e) and Buildings (f) detection based on BPT representation

various nodes within the BPT [35]. Results of this object-oriented pruning are presented on Fig. 4(e)(f) for the detection of roads and buildings, respectively. These results are very promising and outperform standard pixel-wise spectral classification [35]. Gathering connected pixels belonging to meaningful structures into specific nodes, the BPT again turns out to be extremely well suited to the analysis of remote sensing data.

5 Concluding Remarks

Using airborne or satellite platforms, remote sensing is playing a key role in a growing number of applications. The acquired images are more and more complex: increased dimensionality in the case of hyperspectral data, increased complexity of details in the case of images with very high spatial resolution. The structures of interest in these images are of various scales and shapes. In order to tackle the induced issues, multi-scale and hierarchical processing methods are highly desirable. In this context, mathematical morphology provides a set of extremely powerful tools, such as the attribute filters and the binary partition trees. In this paper, we reviewed these strategies, pointing key features regarding their use and extension to remote sensing data. Excellent performances have been achieved on a variety of applications: segmentation, classification or object detection and recognition.

Some open issues remain. We can cite a few of them: when dealing with one specific application, how can one select the optimal set of attributes? Is there a way to achieve scale invariant processing - over what range of scales? - in order to increase the generality of the algorithms? Regarding the spectral dimension: while it is now widely recognized that spatial-spectral approaches dramatically increase the classification performances, the two dimensions (spectral and spatial, respectively) are considered either sequentially or in parallel with a data- or decision fusion step to merge them. Is there a way to take both dimensions into account in a more intricate way? Are the norms and distances designed for the case of hyperspectral data (typically with a few hundreds bands) robust when the dimension drastically increases, as in the case of the ultra-spectral images (typically a few thousands of bands)? Finally, we must underline that this paper was mostly focused on optical data. Images formed using radar sensors (synthetic aperture radar, polarimetric or interferometric data) are also increasingly used and also require new developments. A few papers are already available, but the problematic is quite different as the signal to noise ratio is significantly lower than for optical data and the information about the physical nature and geometry of the actual structures is much more difficult to access. There will undoubtedly be more developments dealing with these data in the coming years.

References

1. Daya Sagar, B.S., Serra, J.: Spatial information retrieval, analysis, reasoning and modelling. International Journal of Remote Sensing 31(22), 5747–5750 (2010)
2. Richards, J.A., Jia, X.: Remote sensing digital image analysis: an introduction. Springer, Heidelberg (2006)
3. Miller, H., Han, J.: Geographic data mining and knowledge discovery. Chapman & Hall/CRC data mining and knowledge discovery series. CRC Press, Boca Raton (2009)

4. Jhung, Y., Swain, P.: Bayesian contextual classification based on modified m-estimates and markov random fields. IEEE Transactions on Geoscience and Remote Sensing 34(1), 67–75 (1996)
5. Datcu, M., Seidel, K., Walessa, M.: Spatial information retrieval from remote-sensing images. i. information theoretical perspective. IEEE Transactions on Geoscience and Remote Sensing 36(5), 1431–1445 (1998)
6. Melgani, F., Serpico, S.: A markov random field approach to spatio-temporal contextual image classification. IEEE Transactions on Geoscience and Remote Sensing 41(11), 2478–2487 (2003)
7. Haralick, R.M., Shanmugam, K., Dinstein, I.H.: Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics 3(6), 610–621 (1973)
8. Jong, S., Meer, F.: Remote sensing image analysis: including the spatial domain. In: Remote Sensing and Digital Image Processing, vol. 1. Kluwer Academic, Dordrecht (2004)
9. Pesaresi, M., Benediktsson, J.A.: A new approach for the morphological segmentation of high-resolution satellite imagery. IEEE Transactions on Geoscience and Remote Sensing 39(2), 309–320 (2001)
10. Bruzzone, L., Carlin, L.: A multilevel context-based system for classification of very high spatial resolution images. IEEE Transactions on Geoscience and Remote Sensing 44, 2587–2600 (2006)
11. Tarabalka, Y., Benediktsson, J., Chanussot, J., Tilton, J.: Multiple spectral-spatial classification approach for hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing 48(11), 4122–4132 (2010)
12. Tarabalka, Y., Benediktsson, J.A., Chanussot, J.: Spectral & spatial classification of hyperspectral imagery based on partitional clustering techniques. IEEE Transactions on Geoscience and Remote Sensing 47(8), 2973–2987 (2009)
13. Tarabalka, Y., Chanussot, J., Benediktsson, J.A.: Segmentation and classification of hyperspectral images using minimum spanning forest grown from automatically selected markers. IEEE Transactions on Systems Man and Cybernetics Part B: Cybernetics 40(5), 1267–1279 (2010)
14. Tarabalka, Y., Chanussot, J., Benediktsson, J.A., Angulo, J., Fauvel, M.: Segmentation and classification of hyperspectral data using watershed. In: Proc. IEEE International Geoscience and Remote Sensing Symposium 2008, IGARSS 2008, July 7-11, vol. 3, pp. III–652–III–655 (2008)
15. Gaetano, R., Scarpa, G., Poggi, G.: Hierarchical texture-based segmentation of multiresolution remote-sensing images. IEEE Transactions on Geoscience and Remote Sensing 47(7), 2129–2141 (2009)
16. Navulur, K.: Multispectral Image Analysis Using the Object-Oriented Paradigm. CRC Press, Inc., Boca Raton (2006)
17. Blaschke, T., Lang, S., Hay, G.: Object-based image analysis: spatial concepts for knowledge-driven remote sensing applications. Lecture notes in geoinformation and cartography. Springer, Heidelberg (2008)
18. Nicolin, B., Gabler, R.: A knowledge-based system for the analysis of aerial images. IEEE Transactions on Geoscience and Remote Sensing GE-25(3), 317–329 (1987)
19. Hay, G.J., Blaschke, T., Marceau, D.J., Bouchard, A.: A comparison of three image-object methods for the multiscale analysis of landscape structure. ISPRS Journal of Photogrammetry and Remote Sensing 57(5-6), 327–345 (2003)
20. Aksoy, S., Koperski, K., Tusk, C., Marchisio, G., Tilton, J.: Learning bayesian classifiers for scene classification with a visual grammar. IEEE Transactions on Geoscience and Remote Sensing 43(3), 581–589 (2005)

21. Serra, J.: Image Analysis and Mathematical Morphology. Theoretical Advances, vol. 2. Academic Press, New York (1988)
22. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, London (1983)
23. Soille, P.: Morphological Image Analysis, Principles and Applications, 2nd edn. Springer, Berlin (2003)
24. Najman, L., Talbot, H.: Mathematical Morphology. Wiley-ISTE (August 2010)
25. Soille, P., Pesaresi, M.: Advances in mathematical morphology applied to geosciences and remote sensing. IEEE Transactions on Geoscience and Remote Sensing 40, 2042–2055 (2002)
26. Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Transactions on Image Processing 4(8), 1153–1160 (1995)
27. Salembier, P.: Connected operators based on region-trees. In: Proc. 15th IEEE International Conference on Image Processing, ICIP 2008, pp. 2176–2179 (2008)
28. Plaza, A., Benediktsson, J., Boardman, J., Brazile, J., Bruzzone, L., Camps-Valls, G., Chanussot, J., Fauvel, M., Gamba, P., Gualtieri, A., Tilton, J., Trianni, G.: Advanced processing of hyperspectral images. Remote Sensing of Environment 113(1), S110–S122 (2009)
29. Gualtieri, J.A., Tilton, J.: Hierarchical segmentation of hyperspectral data. In: AVIRIS Earth Science and Applications Workshop Proceedings, pp. 5–8 (2002)
30. Plaza, A., Tilton, J.: Automated selection of results in hierarchical segmentations of remotely sensed hyperspectral images. In: Proc.of IGARSS 2005 (2005)
31. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Transactions on Image Processing 9(4), 561–576 (2000)
32. Valero, S., Salembier, P., Chanussot, J.: New hyperspectral data representation using binary partition tree. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 80–83 (2010)
33. Valero, S., Salembier, P., Chanussot, J.: Comparison of merging orders and pruning strategies for binary partition tree in hyperspectral data. In: 17th IEEE International Conference on Image Processing (ICIP 2010), pp. 2565–2568 (2010)
34. Valero, S., Salembier, P., Chanussot, J.: Hyperspectral image segmentation using binary partition trees. Submitted to ICIP 2011, Brussels, Belgium (2011)
35. Valero, S., Salembier, P., Chanussot, J., Cuadras, C.: New binary partition tree construction for hyperspectral images: Application to object detection. In: Proc.of IGARSS 2011, Vancouver, Canada (2011)
36. Binaghi, E., Gallo, I., Pepe, M.: A cognitive pyramid for contextual classification of remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 41(12), 2906–2922 (2004)
37. Valero, S., Chanussot, J., Benediktsson, J., Talbot, H., Waske, B.: Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images. Pattern Recognition Letters 31(10), 1120–1127 (2010)
38. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. Comput. Vis. Image Underst. 64(3), 377–389 (1996)
39. Dalla Mura, M., Benediktsson, J.A., Waske, B., Bruzzone, L.: Morphological attribute filters for the analysis of very high resolution remote sensing images. In: Proc. IEEE International Geoscience and Remote Sensing Symposium 2009, IGARSS 2009, vol. 3, pp. III–97–III–100 (July 2009)

40. Dalla Mura, M., Benediktsson, J.A., Waske, B., Bruzzone, L.: Morphological attribute profiles for the analysis of very high resolution images. IEEE Transactions on Geoscience and Remote Sensing 48(10), 3747–3762 (2010)
41. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. IEEE Transactions on Image Processing 7(4), 555–570 (1998)
42. Monasse, P., Guichard, F.: Fast computation of a contrast-invariant image representation. IEEE Transactions on Image Processing 9(5), 860–872 (2000)
43. Maragos, P., Ziff, R.: Threshold superposition in morphological image analysis systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(5), 498–504 (1990)
44. Dalla Mura, M., Benediktsson, J.A., Waske, B., Bruzzone, L.: Extended profiles with morphological attribute filters for the analysis of hyperspectral data. International Journal of Remote Sensing 31(22), 5975–5991 (2010)
45. Alonso-Gonzalez, A., Lopez-Martinez, C., Salembier, P.: Filtering and segmentation of polarimetric SAR images with binary partition trees. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2010), pp. 4043–4046 (2010)
46. Dalla Mura, M., Benediktsson, B., Bruzzone, L.: Self-dual attribute profiles for the analysis of remote sensing images. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 306–319. Springer, Heidelberg (2011)
47. Wilkinson, M.H.F., Gao, H., Hesselink, W.H., Jonker, J.E., Meijster, A.: Concurrent computation of attribute filters on shared memory parallel machines. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(10), 1800–1813 (2008)
48. Dalla Mura, M., Benediktsson, J., Chanussot, J., Bruzzone, L.: The Evolution of the Morphological Profile: from Panchromatic to Hyperspectral Images. In: Optical Remote Sensing - Advances in Signal Processing and Exploitation Techniques. Springer, Heidelberg (2011)
49. Benediktsson, J.A., Palmason, J.A., Sveinsson, J.R.: Classification of hyperspectral data from urban areas based on extended morphological profiles. IEEE Transactions on Geoscience and Remote Sensing 43(3), 480–491 (2005)
50. Falco, N., Dalla Mura, M., Bovolo, F., Benediktsson, J.A., Bruzzone, L.: Study on the capabilities of morphological attribute profiles in change detection on VHR images. In: Bruzzone, L. (ed.) Image and Signal Processing for Remote Sensing XVI. Proceedings of SPIE, vol. 7830. SPIE, Bellingham (2010)
51. Alonso-Gonzalez, A., Lopez-Martinez, C., Salembier, P.: Filtering and segmentation of polarimetric sar images with binary partition trees. In: Proc. IEEE International Geoscience and Remote Sensing Symposium 2010, IGARSS 2010, Honolulu, USA, pp. 4043–4046 (2010)
52. Calderero, F., Marques, F.: Region merging techniques using information theory statistical measures. IEEE Trans. Image Processing 19, 1567–1586 (2010)
53. Hu, M.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory 8(2), 179–187 (1962)
54. Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001)
55. Cardoso, J., Corte-Real, L.: Toward a generic evaluation of image segmentation. IEEE Trans. Image Processing 14, 1773–1782 (2005)

Self-dual Attribute Profiles for the Analysis of Remote Sensing Images

Mauro Dalla Mura[1,2], Jon Atli Benediktsson[2], and Lorenzo Bruzzone[1]

[1] Department of Information Engineering and Computer Science,
University of Trento, Via Sommarive 14 I-38123, Povo, Trento, Italy
dallamura@disi.unitn.it, bruzzone@ing.unitn.it
[2] Faculty of Electrical and Computer Engineering, University of Iceland,
Hjardarhaga 2-6, 101 Reykjavik, Iceland
benedikt@hi.is

Abstract. The spatial relations are essential information that should be considered when analyzing remote sensing images. Attribute profiles (combinations of an anti-granulometry and a granulometry computed with connected operators based on attributes) can be employed for the modeling of the spatial information of the surveyed scene. In this paper we propose self-dual attribute profiles which are attribute profiles computed on an inclusion tree with self-dual operators. The proposed variant of the attribute profile was effectively considered for the classification of a very high geometrical resolution remote sensing image.

1 Introduction

In general, spatial features, such as borders, edges, discontinuities, surfaces, shapes, etc. encode most of the semantic of the objects represented in an image. Thus, spatial features are ones of the most relevant characteristics for image understanding due to their perceptual relevance. With specific attention to the remote sensing field, the spatial characteristics are probably the most informative features when dealing with images of very high geometrical resolution (VHR). Furthermore, if we consider images acquired by spaceborne optical passive sensors, a dense sampling of the surveyed scene in the spatial domain (leading to images with high geometrical resolution) limits the radiometric and spectral resolution of the acquired data due to the finite and relative low energy emitted by the surface and captured by the sensor. In fact, to a high geometrical resolution corresponds in general low radiometric and spectral resolutions and usually VHR images are composed by only one spectral band (i.e., panchromatic bands). In this scenario, the analysis of VHR imagery has to heavily rely on the spatial characteristics since they embed most of the semantic information.

Approaches based on mathematical morphology have proved to be an effective set of tools for including spatial information in the analysis of remote sensing images [1]. In particular, connected operators [2] have shown to be appropriate for the processing of VHR remote sensing images since they do not introduce new discontinuities in the image (i.e., the borders of the objects in the scene

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 320–330, 2011.

are not distorted) [1,3,4,5,6,7]. In [3] Morphological Profiles (MPs) were introduced for modeling the spatial information of the scene for segmentation. An MP is a concatenation of an anti-granulometry and a granulometry computed with connected operators such as opening and closing by reconstruction with structuring element of fixed shape (in general square or disk) and increasing size. Basically, the MP produces copies of the original image with an increasing level of simplification obtained by merging progressively larger flat regions performing a multiscale decomposition of the image. Applications of morphological operators in a multiscale approach date back to pattern spectrum [8]. The characterization of the spatial information obtained by the application of a MP is particularly suitable for representing the multi-scale variability of the structures in the image but it is not sufficient to model other geometrical features (e.g., shape descriptors). To avoid this limitation, the use of connected operators based on attributes computed on the regions of the image (i.e., attribute filters [9]) instead of the conventional operators based on a structuring element and the geodesic reconstruction was proposed in [4] leading to the definition of Attribute Profiles (APs) as a generalization of the concept of MP. The application of attribute filters in a multilevel architecture was also presented in [10]. The AP, compared to an MP, is characterized by a greater flexibility on the definition of the connected operator employed (i.e., due to the freedom in the selection of the attributes), which permits to model other geometrical characteristics rather than the size of the objects. Moreover, APs result also less demanding in terms of computational complexity since they are based on a max-tree and min-tree for the computation of a granulometry and an anti-granulometry, respectively. Min- and max-tree are efficient hierarchical tree representations of the image [11,2]. The APs showed also interesting characteristics when extended to hyperspectral images [6]. In greater details, analogously to [5], the APs were applied to the first principal components extracted from a hyperspectral image, generating an Extended Attribute Profile (EAP).

Since the APs are based on the application of either extensive or anti-extensive operators, the multilevel simplification is either obtained on the bright or dark components of the image. If a simultaneous simplification of bright and dark regions is aimed, self-dual operators should be used. The application of self-dual connected operators leads to an image simplification characterized by more homogeneous regions with respect to the results obtained by extensive or anti-extensive connected operators. In a remote sensing scene this effect can be useful for flattening textured areas (e.g., agricoltural fields, vegetated areas, etc.) or removing both dark and bright details. In [7] we investigated the effects of Alternating Sequential Attribute Filters (ASAFs) obtained by the application of attribute thinning and attribute thickening in an alternating sequential approach on a VHR remote sensing image. The selection of different attributes with progressively stricter criteria (i.e., producing greater simplifications of the image) showed how significantly different effects can be obtained on the image. However, the alternating sequence of thinning and thickening operators is not a self-dual operator since the filtering effect is biased by the operator that is applied first.

Furthermore, the non approximated computation of ASAFs is computationally very demanding.

In this paper we propose to use an inclusion tree for computing the AP instead of the min-tree and max-tree as done in [4]. The inclusion tree is a tree representation of an image which fuses both the min-tree and max-tree of the image in a single data structure [12]. The main advantages for the use of an inclusion tree for the computation of a profile relies on: i) the construction and subsequent manipulation of a single representation of the image embedding the min- and max-tree representation (requiring less resources in term of computational complexity and memory occupation); and ii) the capability of computing extensive, anti-extensive or self-dual connected operators. In this paper we focus on the latter aspect investigating Self-Dual Attribute Profiles (SDAPs) which are APs based on self-dual connected operators for the classification of a VHR remote sensing image.

The paper is divided into five sections. The next section recalls some background notions on the representation of the image as hierarchical tree of flat regions. Section 3 presents the definition of the SDAPs. The experimental results are reported in Section 4. Finally, the conclusions are drawn in Section 5.

2 Representations of an Image as a Tree

Let us consider a discrete 2D image u that is a map from the discrete image domain $E \subseteq \mathbb{Z}^2$ to $\mathbb{Z}$. Since the codomain of u is ordered, the image can be fully represented by its *upper* or *lower level sets*, which are defined as

$$[u \geq \lambda] = \{x \in E, u(x) \geq \lambda\}, \quad [u < \lambda] = \{x \in E, u(x) < \lambda\}, \quad \lambda \in \mathbb{Z}.$$

Thus, upper and lower level sets are composed by binary images obtained by thresholding the input image at all the values mapped by the function u. The connected components extracted by the binarization of the input image related to the upper or lower level sets can be grouped in the sets:

$$\mathcal{U}(u) = \{X : X \in \mathcal{CC}([u \geq \lambda]), \lambda \in \mathbb{Z}\} \quad \mathcal{L}(u) = \{X : X \in \mathcal{CC}([u < \lambda]), \lambda \in \mathbb{Z}\}$$

with $\mathcal{CC}(f)$ the connected components of the generic image f. If we consider u as a function of the height (coded by the values of the codomain), the upper (lower) level sets are obtained by slicing the topographic relief at different heights and projecting the points with greater (lower) values than the thresholding height to the plane at the given height. By varying the height of the plane, it can be seen how connected components (i.e., regions of isolevel) can merge, enlarge, shrink, split, appear or disappear according to the morphology of the elevation surface. Among the connected components extracted by either the upper or lower level sets (belonging to $\mathcal{U}(u)$ or $\mathcal{L}(u)$ respectively) there is an inclusion relationship [12]. In greater details, any two components $A, B \in \mathcal{U}(u)$ are either nested ($A \subseteq B$ or $B \subseteq A$) or disjoint ($A \cap B = \emptyset$). Analogous considerations can be done for $\mathcal{L}(u)$. Due to the inclusion relations between the flat zones in the image, it is possible to associate a node of a tree to each connected component and represent the image as a hierarchical structure. The hierarchical tree representing the components in $\mathcal{U}(u)$ ($\mathcal{L}(u)$) and their inclusion relations is called *max-tree* (*min-tree*) [11]. In the max-tree representation, the root node is the entire image domain at the lowest grayscale value while the leaves of the tree are the regional maxima. By duality, the min-tree can be obtained by generating the max-tree on the complement of the image and shows as root the whole image at the highest grayscale value and as leaves the regional minima. Both min- and max-trees are equivalent representations of the image. However, not all the components present in $\mathcal{U}(u)$ are also present in $\mathcal{L}(u)$ and vice versa. For example, the top of two peaks (i.e., regional maxima) in the image having the same height, will be represented in the upper level set as two distinct components while in the lower level set they will belong to the same component. A self-dual representation of the connected components of an image called *inclusion tree* (or *tree of shapes*) exists for continuous images [13]. When dealing with discrete images (as in this work) different discrete connectivity rules have to be associated to minima and maxima regions (e.g., 8- and 4-connectivity for the definition of the upper and lower level sets, respectively [12]) leading to a quasi self-dual representation of the image. If only one type of connectivity were used for computing the inclusion tree, inconsistent results would be obtained since the notion of hole is not properly defined. We believe that the fact that the inclusion tree is not completely self-dual does not affect the results for practical applications. For example, even connected operators applied in an alternating sequence (providing a quasi self-dual effect biased by which operator starts first in the sequence) proved to be suitable for obtaining a simplification of the image [7]. The inclusion tree is also a complete representation of the input image (i.e., it is possible to fully retrieve the image from the tree). The inclusion relations between flat regions represented in this tree structure is given by the *saturation* of the connected components. The saturation is an operator that fills the *holes* of a component. With holes of a connected component A are intended all those regions that belong to the background of A but which are not connected to its border. A saturated region (i.e., component with its holes filled) is also called a *shape*. Thus, according to the operator of saturation, a component A can be considered as included in B (regardless their relative graylevel difference) if $A \subseteq B$ in $\mathcal{U}(u)$ or $\mathcal{L}(u)$. The construction of the tree can be performed by an efficient algorithm called Fast Level Set Transform (FLST) that starts from considering the regional extrema (leaves of the tree) and progressively saturates the components until only a single flat region (the root of the tree) is obtained [13]. The progressive saturation of the components explains the inclusion relations on which the tree is constructed. The inclusion tree is a more general representation of the image with respect the max- and min-trees since comprehends both the $\mathcal{U}(u)$ and $\mathcal{L}(u)$ sets. Another representation of the image as hierarchical tree of isolevel regions is given by *binary partition trees* (BPTs) [14]. Since BPTs contain those connected components considered more interesting according to the criterion involved during the tree creation (e.g., homogeneity, size, contrast, etc.) they can enhance certain features (according to the criterion used) more than a min- max- or inclusion tree. Connected operators can be efficiently computed on the trees generated from the image [2]. The representation of the image as max-tree is useful for performing anti-extensive connected operators (e.g., thinning). By duality, on the min-tree can be computed extensive connected operators (e.g., thickening). Self-dual connected operators can be obtained by considering an inclusion tree. Moreover, since the inclusion tree stores both the components of the upper and lower level sets, one can perform an anti-extensive or extensive transformation by only considering the regions of the upper or lower level set, respectively.

Since connected operators modify an image only by merging its flat zones, the filtering is performed on the hierarchical tree as a pruning. In general, the tree is pruned by evaluating a binary predicate T on the nodes and removing those that do not fulfill it. The predicate usually compares an attribute *attr* computed on the pixels belonging to a connected component C (corresponding to a node in the tree) and a given threshold value taken as reference λ: e.g., $T = attr(C) \geq \lambda$. The attributes can be any measure computable on the regions (e.g., area, volume, entropy, etc.).

If we consider a filtering done according to an increasing predicate (i.e., for the connected components A and B holds that if $T(A) = true$ then $T(B) = true$ with $B \subseteq A$) the connected operator obtained will also be increasing and the pruning can be performed by removing entire branches (constituted by a node and all its descendants) from the tree. If the predicate is non-increasing, then different filtering rules can be applied in order to determine which nodes have to be removed since the evaluated criterion could be fulfilled for certain nodes but not for their descendants [2].

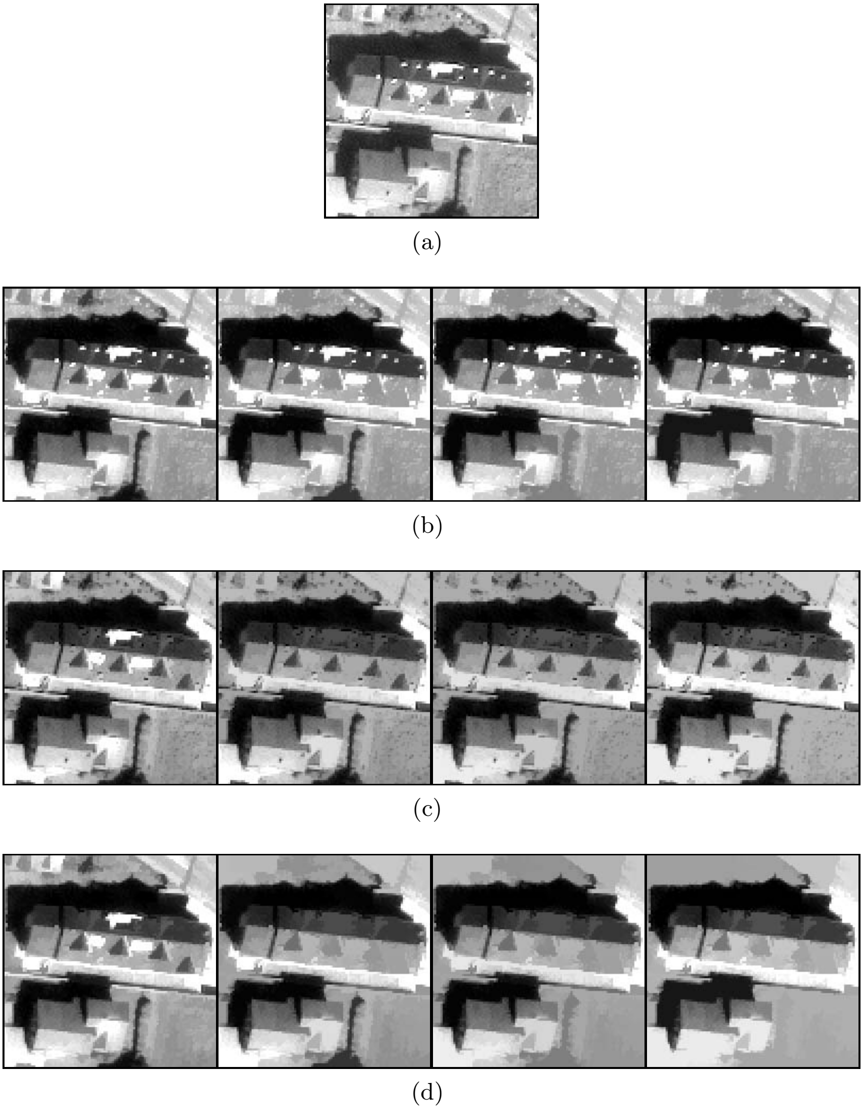
3 Self-dual Attribute Profiles

As a generalization of the concept of MP [3], the AP of an image u was defined in [4] as:

$$AP(u) = \left\{ \phi^{T_{\lambda_L}}(u), \phi^{T_{\lambda_{L-1}}}(u), \ldots, \phi^{T_{\lambda_1}}(u), u, \gamma^{T_{\lambda_1}}(u), \ldots, \gamma^{T_{\lambda_{L-1}}}(u), \gamma^{T_{\lambda_L}}(u) \right\}, \tag{1}$$

with $\{T_\lambda\}$ a set of L ordered predicates (i.e., $T_{\lambda_i} \subseteq T_{\lambda_k}$, $i \leq k$), and ϕ, γ thickening and thinning operators, respectively. If we take into account the inclusion tree, the AP can be obtained by considering in the filtering the components belonging to the upper level set or lower level set for performing a thinning or thickening, respectively. Moreover, the phase devoted to the construction of the inclusion tree with the FLST implementation results faster and requires less memory for storing the data than the construction of both a min- and max-tree [12]. The use of an inclusion tree permits also to filter the image with self-dual operators when the inclusion relations defined by the saturation of the components are considered instead of those belonging to the upper or lower level set. Self-dual operators are advisable for the processing of remote sensing images when a simplification on both bright and dark structures is required.

As an example of this dual effect, Fig. 1 shows the results obtained by filtering a particular of a VHR remote sensing panchromatic image considering as a predicate $T = card(C) \geq \lambda$ with $card(C)$ the cardinality of the connected component C with increasing values of λ. The filtering was computed on a


(a)

(b)

(c)

(d)

Fig. 1. Particular of the panchromatic band of 0.5 [m] resolution acquired by GeoEye-1 image of Borgo Valsugana, Trento, Italy reported in Fig. 2(a). (a): Original image; (b): Area closing (ϕ^{T_λ}); (c): Area opening (γ^{T_λ}); (d): Grain filter (ρ^{T_λ}). The values of area taken as reference and used by all the three operators are $\lambda = \{50, 500, 1000, 2000\}$ (correspondent to images from left to right).

min-, max- and inclusion tree. The operator obtained with the considered predicate on a max-tree (min-tree) was basically an area opening (area closing). When considering the inclusion tree, a self-dual operator (that is called grain filter in [12]) was applied. The latter operator could be seen as a connected opening in the shape-tree semilattice [15,16]. By analyzing the figure, it is possible to see how bright and dark details were preserved unaffected in the images filtered with area closing and opening, respectively. For example the bright areas on the roof (probably due to glares of roof windows, metal plates or solar cells) of the building on the top of the image and bright small regions composing the texture of the garden on the bottom right were completely preserved by the area closing. Analogously, the shadows casted on the roof by the dormers and other shaded regions on the vegetated area on the top of the image were unaffected by all the openings. In comparison, the effect of the self-dual filter can be noticed in the production of more homogeneous regions since both bright and dark components were simultaneously filtered. For instance, the particulars on the roofs were removed and the textured areas were completely flattened.

Analogously to the definition of the AP (1), it is possible to derive a formulation of Self-Dual Attribute Profiles (i.e., APs built with self-dual operators):

$$SDAP(u) = \left\{u, \rho^{T_{\lambda_1}}(u), \ldots, \rho^{T_{\lambda_{L-1}}}(u), \rho^{T_{\lambda_L}}(u)\right\}, \tag{2}$$

with ρ the self-dual operator based on the predicate T, and being $\{T_\lambda\}$ a set of L ordered predicates. In contrast to APs, the SDAP is composed by $L+1$ images while the AP built with the same sequence of λs is made up of $2L+1$ images.

4 Experimental Analysis

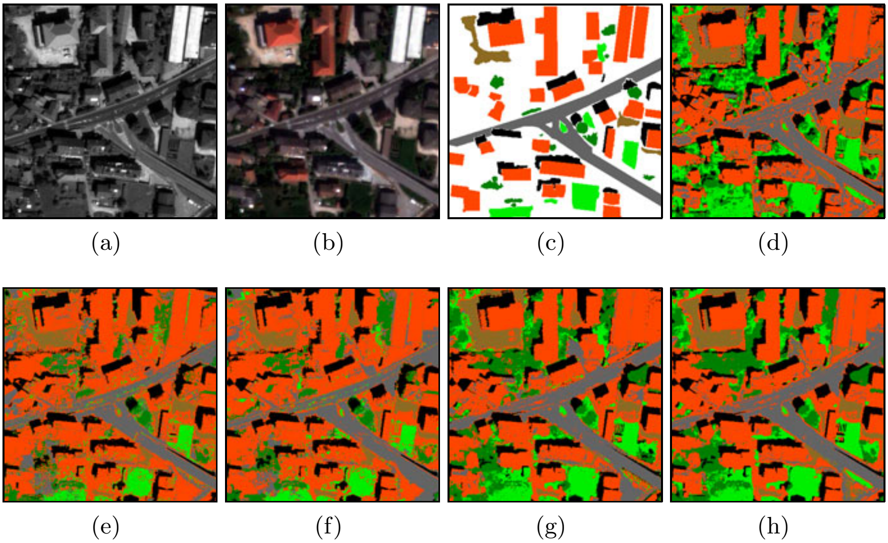
In the experimental analysis, a VHR image acquired by GeoEye-1 over Borgo Valsugana, a village close to the city of Trento, Italy, was considered for classification. The data set is composed by a panchromatic band and four multispectral (MS) pansharpened images (acquired on the visible and near infrared electromagnetic spectrum) of 400×400 pixels with a geometrical resolution of 0.5 m. The Normalized Vegetation Index (NDVI) image was also generated for enhancing the vegetated areas. The NDVI is given by $\frac{NIR-R}{NIR+R}$ with NIR and R the bands acquired on the near infrared and red regions, respectively. The scene presents heterogeneous residential buildings, roads and some vegetated areas. Six thematic classes were identified in the image: Buildings, Roads, Trees, Meadows, Shadows and Soil. A reference map of the coverage classes was generated by visual inspection leading to a total of 67977 labeled pixels.

For including the spatial information in the analysis, an AP and a SDAP with area attribute and 12 reference values (λ = {5, 25, 50, 100, 150, 200, 300, 500, 750, 1000, 2000, 3000 }) were computed on the panchromatic image. The AP was computed using the implementation of the min and max-tree included in the C++ Milena library [17] and the SDAP from an adaptation of the code for the inclusion tree provided in the MegaWave2 toolbox[1].

[1] Available at `http://megawave.cmla.ens-cachan.fr`

The data set was classified by 6 different classifiers: Linear Bayes Normal classifier, Quadratic Bayes Normal classifier, Random Forest (RF), 3-Nearest Neighbor, SVM with linear kernel and SVM with RBF kernel. Part of the labeled samples of the reference image was considered for the training of the classifiers, the rest was used for computing the classification accuracy, which was assessed by the Overall Accuracy (OA) and the Kappa coefficient (κ). Two training sets were considered taking for all the classes a total of the 1% (685) and 10% (6801) of the reference samples.

The classification accuracies obtained by considering different features are reported in Table 1. With 10% of the samples used for training, it can be seen that considering the images of the SDAP as features and the RF as classifier outperformed in terms of overall accuracies the best results obtained by the spectral features (MS + NDVI) and AP taken singularly of 0.25% and 2.34%, respectively. By a visual inspection of the map correspondent to the best accuracy obtained by the spectral features among all the classifiers (Fig. 2d) is possible to notice that the vegetation was well classified (also separating meadows from trees). However, roads and buildings were often mixed and the shapes of some


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

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

Fig. 2. GeoEye-1 Borgo Valsugana data set. (a) Panchromatic band; (b) True color composition of the pansharpened multispectral channels; (c) Map of the reference samples. Classification maps (all taking the 10% of the reference samples as training) obtained by: (d) MS + NDVI and SVM with RBF kernel classifier (OA 86.23%); (e) AP and RF classifier (OA 84.14%); (f) SDAP and RF classifier (OA 86.48%); (g) MS + NDVI + AP and SVM with RBF kernel classifier (OA 93.45%); (h) MS + NDVI + SDAP and RF classifier (OA 93.50%). Thematic classes: 🟥 buildings, ▨ roads, 🟩 trees, 🟩 meadows, ⬛ shadows, 🟫 soil.

objects were distorted (see the shadow of the building on the top left). In contrast to Fig. 2d, the maps obtained by the AP (Fig. 2e) and SDAP (Fig. 2f) show less confusion between roads and buildings but the natural classes (i.e., soil, meadows and trees) were not correctly detected. Furthermore, Fig. 2f in comparison to Fig. 2e shows more homogeneous regions (due to the use of self-dual operators in the computation of the profile). When the spectral features are considered along with the AP or SDAP, the best accuracies among all the experiments were obtained. In particular, considering the training set as 10% and the best obtained OAs, there was an increase of accuracy up to about 7% compared to considering both the spectral features and the SDAP singularly and about 9% with respect to considering only the AP. A greater precision in detecting buildings, roads and vegetation is also clear from the maps (Fig. 2g, 2h). When comparing the best overall accuracies obtained by considering the AP against the SDAP, both with the spectral features, similar results were obtained (with a SVM with RBF kernel and a RF, respectively). However, the results obtained with the SDAP slightly outperformed those achieved with the AP. Again, when comparing Fig. 2g and 2h, the use of SDAP produced a less noisy map than considering the AP. However, in Fig. 2h some shadows in the vegetated areas were not correctly detected compared to Fig. 2g. Satisfactory results are also obtained with a reduced training set (1% of the reference samples) confirming the improvements given by using the SDAP.

Table 1. GeoEye-1 Borgo Valsugana data set. Classification accuracies obtained by classifying conbinations of the spectral features (MS + NDVI), the AP and SDAP with a linear (Lin.), quadratic (Quad.), random forest (RF), 3-nearest neighbor (3-NN), SVM with linear kernel (SVM Lin.) and SVM with RBF kernel (SVM RBF) classifiers. The best accuracies for each classifier and among all the experiments are marked in bold and underlined, respectively.

	Lin.		**Quadr.**		**RF**		**3-NN**		**SVM Lin.**		**SVM RBF**	
Train	1%	10%	1%	10%	1%	10%	1%	10%	1%	10%	1%	10%
	MS + NDVI (5 feats)											
OA (%)	63.12	63.90	73.12	73.03	81.50	85.58	80.57	83.69	69.03	69.25	82.16	86.23
κ (%)	45.72	46.47	63.22	63.27	73.20	79.37	72.27	76.77	51.19	51.21	74.50	80.41
	AP (25 feats)											
OA (%)	62.23	63.34	37.45	22.83	77.35	84.14	71.23	80.29	66.22	67.94	76.15	83.71
κ (%)	41.44	43.12	24.28	12.64	66.85	77.05	58.24	71.55	51.98	51.96	65.91	76.43
	SDAP (13 feats)											
OA (%)	62.97	64.07	54.60	52.36	80.36	86.48	75.27	83.75	61.35	63.25	78.77	83.96
κ (%)	42.37	43.29	42.54	41.64	71.72	80.50	64.08	76.35	33.95	38.18	69.41	77.44
	MS + NDVI + AP (30 feats)											
OA (%)	**77.85**	**78.86**	44.30	29.40	87.57	92.58	84.66	90.51	**85.95**	**88.68**	**88.53**	**93.45**
κ (%)	**68.54**	**70.26**	30.30	17.62	82.06	89.37	78.19	86.50	**80.30**	**84.13**	**83.71**	**90.68**
	MS + NDVI + SDAP (18 feats)											
OA (%)	74.99	76.00	**75.34**	**73.75**	**89.31**	**93.50**	**85.47**	**90.62**	80.04	80.28	88.20	93.33
κ (%)	64.33	65.82	**66.17**	**65.03**	**84.65**	**90.70**	**79.35**	**86.65**	71.06	71.04	83.27	90.48

5 Conclusions

In this paper we have proposed to compute Attribute Profiles on the inclusion tree of the image instead of considering a min- and max-tree. The use of the inclusion tree as structure representing the image contains the information of both the min- and max-tree. Moreover, on the inclusion tree can also be computed self-dual connected operators, which produce a greater simplification of the image with respect to non dual filters since they operate simultaneously on the bright and dark components of the image. Thus, we have derived the definition of Self-Dual Attribute Profiles as a version of the APs based on self-dual operators.

In the experimental analysis carried out, we considered for classification a remote sensing image acquired by GeoEye-1 with geometric resolution of 0.5 m on an area close to Trento, Italy. An AP and a SDAP were computed on the panchromatic band with area attribute and same values taken as reference in the computation of the profile. The results obtained showed how including the AP or SDAP as features in the classification of the spectral features greatly improves the accuracies with respect to considering only the spectral information. The use of the SDAP against the AP leads to better results in terms of accuracies in most of the cases (also with a reduced training set). The greater accuracies were also supported by the obtained maps showing regions classified more homogeneously. The best overall accuracies among all the experiments were obtained by the spectral features and the SDAP with a random forest classifier. The obtained results proved that the use of the SDAP is effective for modeling the spatial information of the scene even with a reduced number of features with respect to considering the AP.

References

1. Soille, P., Pesaresi, M.: Advances in mathematical morphology applied to geosciences and remote sensing. IEEE Trans. Geosci. Rem. Sens. 40, 2042–2055 (2002)
2. Salembier, P., Wilkinson, M.H.F.: Connected operators. IEEE Signal Process. Mag. 26(6), 136–157 (2009), doi:10.1109/MSP.2009.934154
3. Pesaresi, M., Benediktsson, J.A.: A new approach for the morphological segmentation of high-resolution satellite imagery. IEEE Trans. Geosci. Rem. Sens. 39(2), 309–320 (2001), doi:10.1109/36.905239
4. Dalla Mura, M., Benediktsson, J.A., Waske, B., Bruzzone, L.: Morphological attribute profiles for the analysis of very high resolution images. IEEE Trans. Geosci. Rem. Sens. 48(10), 3747–3762 (2010), doi:10.1109/TGRS.2010.2048116
5. Benediktsson, J.A., Palmason, J.A., Sveinsson, J.R.: Classification of hyperspectral data from urban areas based on extended morphological profiles. IEEE Trans. Geosci. Rem. Sens. 43(3), 480–491 (2005), doi:10.1109/TGRS.2004.842478
6. Dalla Mura, M., Benediktsson, J.A., Waske, B., Bruzzone, L.: Extended profiles with morphological attribute filters for the analysis of hyperspectral data. Int. J. Remote Sens. 31(22), 5975–5991 (2010)
7. Dalla Mura, M., Benediktsson, J.A., Bruzzone, L.: Alternating sequential filters with morphological attribute operators for the analysis of remote sensing images. In: Proceedings of SPIE, vol. 7830 (2010)

8. Maragos, P.: Pattern spectrum and multiscale shape representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(7), 701–716 (1989)
9. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. Comput. Vis. Image Underst. 64(3), 377–389 (1996)
10. Urbach, E.R., Roerdink, J.B.T.M., Wilkinson, M.H.F.: 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), 272–285 (2007), doi:10.1109/TPAMI.2007.28
11. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. IEEE Trans. Image Process. 7(4), 555–570 (1998), doi:10.1109/83.663500
12. Caselles, V., Monasse, P.: Geometric Description of Images as Topographic Maps. Springer, Heidelberg (2010)
13. Monasse, P., Guichard, F.: Fast computation of a contrast-invariant image representation. IEEE Trans. Image Process. 9(5), 860–872 (2000), doi:10.1109/83.841532
14. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Trans. Image Process. 9(4), 561–576 (2000), doi:10.1109/83.841934
15. Keshet, R.: Shape-tree semilattice. Journal of Mathematical Imaging and Vision 22(2), 309–331 (2005)
16. Keshet, R.: Adjacency lattices and shape-tree semilattices. Image and Vision Computing 25(4), 436–446 (2007)
17. Levillain, R., Géraud, T., Najman, L.: Milena: Write generic morphological algorithms once, run on many kinds of images. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 295–306. Springer, Heidelberg (2009)

Concurrent Computation of Differential Morphological Profiles on Giga-Pixel Images

Michael H.F. Wilkinson[1], Pierre Soille[2],
Martino Pesaresi[2], and Georgios K. Ouzounis[2]

[1] Johann Bernoulli Institute, University of Groningen, The Netherlands
m.h.f.wilkinson@rug.nl

[2] Geo-Spatial Information Analysis for Global Security and Stability,
Global Security and Crisis Management Unit,
Institute for the Protection and Security of the Citizen,
Joint Research Centre, European Commission, Ispra, Italy
{georgios.ouzounis,pierre.soille,martino.pesaresi}@jrc.ec.europa.eu

Abstract. In this paper we provide an efficient parallel algorithm for reconstruction from markers, and multi-scale analysis through differential morphological profiles, which are top-hat scale spaces based on openings and closings by reconstruction. The new algorithms provide speed gain in two ways: (i) through parallelism, and (ii) through more efficient re-use of previously computed data. The best version of the algorithm provided a 17× speed-up on 24 cores, over computation of the same algorithm on a single core. Compared to the basic method of repeated reconstructions by a sequential algorithm, a speed gain of 25.1 times was obtained.

1 Introduction

The Differential Morphological Profile (DMP) [3,2] is a feature descriptor that often finds usage in the classification and categorisation of man-made structures in very high resolution (VHR) satellite image analysis. It is based on multiple computations of openings and closings by reconstruction, at an increasing sequence of scales. Due to the large data bulk acquired and the need for rapid analysis in cases of disaster response, parallel computation is essential, but is not readily achieved [7]. In this paper we provide a parallel algorithm for the DMP, based on the algorithm for attribute filters in [8], which is easily extended to multi-scale computation. The contribution is twofold: (i) we parallelize reconstruction from markers, and (ii) we increase speed by building and interim data structure known as the Max-Tree [4] once, and compute multiple reconstructions from it, extending the work in [9]. The concurrent implementation presented, tested on a 24-core system, achieves a speed-up of 17, and total speed increase of 25.1 times compared to the straightforward computation on a single core using repeated, sequential reconstruction.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 331–342, 2011.

2 Differential Morphological Profiles

Let E be the definition domain of an arbitrary image f. The differential morphological profile (DMP) [3] of a point $x \in E$, is the concatenation of two vectors that are perpendicular to the image plane, and in opposite direction with respect to each other. They are called the differential opening and closing profiles of x, each consisting of $(I-1)$ elements, where I is the number of scales.

Given an $(I-1)$-scales top-hat and bottom-hat scale space based on openings and closings by reconstruction, the differential opening profile is given by:

$$\Delta^{\Pi}(\gamma^{\rho}(f))(x) = \Big(\big(\gamma^{\rho}_{\lambda_{i-1}}(f) - \gamma^{\rho}_{\lambda_i}(f)\big)(x) \mid \lambda_i > \lambda_{i-1},\ \forall\, i \in [0, 1, ..., I-1]\Big), \quad (1)$$

in which $\gamma^{\rho}_{\lambda_i}$ is the opening by reconstruction with a disk of radius λ_i and $\gamma^{\rho}_{\lambda_0}(f) = f$. The differential closing profile $\Delta^{\Pi}(\varphi^{\rho}(f))(x)$ is defined analogously. The differential morphological profile DMP of a point x is the $2(I-1)$ long concatenation (denoted by $\sqcup$) of $\Delta^{\Pi}(\gamma^{\rho}(f))(x)$ and $\Delta^{\Pi}(\varphi^{\rho}(f))(x)$, i.e.:

$$\mathrm{DMP}(x) = \Delta^{\Pi}(\gamma^{\rho}(f))(x) \sqcup \Delta^{\Pi}(\varphi^{\rho}(f))(x). \quad (2)$$

The set of DMPs for the entire E is referred to as the DMP vector field and the two scale spaces, as the *opening instance* and *closing instance* of the DMP respectively. An example DMP is shown in Fig. 1. The image on the top of (a) is a tile extract from the panchromatic channel of a Quickbird Imagery © DigitalGlobe Inc., 2007, distributed by Eurimage. It shows the military museum of Istanbul, Turkey. The DMP vector field has also been used to define a segmentation scheme based on the morphological characteristics as in [3], which is often used for the extraction of build-up in urban areas.

Let the *multi-scale opening characteristic* of a DMP at point $x \in E$ be the value :

$$\mathrm{MOC}(x) = \wedge \Big\{ \lambda \mid \Delta(\gamma^{\rho}_{\lambda}(f))(x) = \vee \Delta^{\Pi}(\gamma^{\rho}(f))(x) \Big\}. \quad (3)$$

Equation (3) states that MOC is the smallest scale λ associated to the maximal intensity difference of the DMP vector field at point x. The *multi-scale closing characteristic* (MCC) is the complimentary scale space, defined either as the MOC of the inverted image $-f(x)$, or equivalently from the closing-by-reconstruction top-hat scale space. The multi-scale leveling-like segmentation (MSLS) into convex/concave/flat regions [3] is then defined as:

$$s(x) = \begin{cases} \mathrm{MOC}(x) & \text{if } \vee \Delta^{\Pi}(\gamma^{\rho}(f))(x) > \vee \Delta^{\Pi}(\varphi^{\rho}(f))(x), \quad (4a) \\ \mathrm{MCC}(x) & \text{if } \vee \Delta^{\Pi}(\gamma^{\rho}(f))(x) < \vee \Delta^{\Pi}(\varphi^{\rho}(f))(x), \quad (4b) \\ 0 & \text{otherwise.} \quad (4c) \end{cases}$$

Note that instead of assigning the scale parameter λ on each output pixel, the segmentation is often given by a three level output; 255 in case of convexity, 0 in case of concavity and 128 in case of flatness. This segmentation scheme is often used for the extraction of build-up in urban areas. As this is a point operation, it is trivial to parallelize. Therefore, we did not time this particular phase of the analysis.

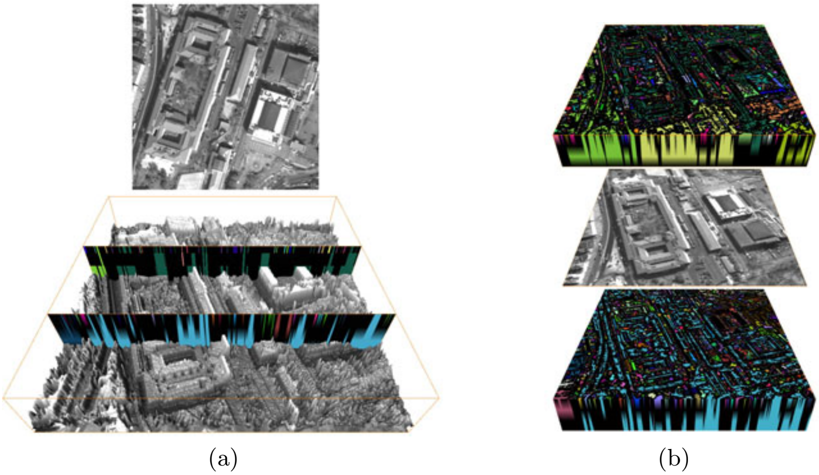


Fig. 1. DMP vector field example: the original image tile at the top of (a), its 3-scale DMP vector field (b) The bottom image in (a) shows the intensity field of the original with two cross-sections; the first one to the viewer corresponding to the closing and the second to the opening instance of the DMP vector field.

3 Parallel Computation

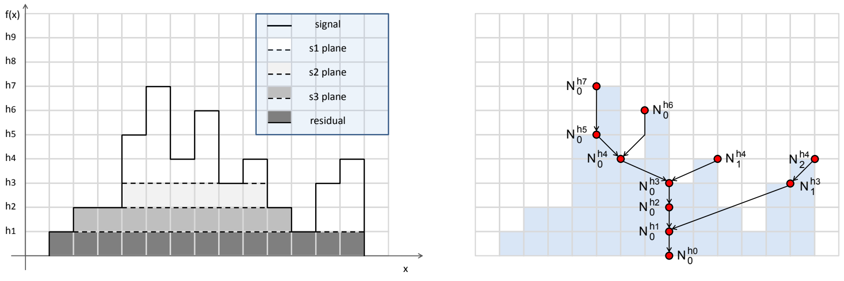
For the opening part of the DMP we first compute the marker images, by erosions with exact Euclidean disks with the algorithm from [5]. This is parallelized trivially by splitting the image into as many slices as there are processors, and the algorithm is applied to each slice separately.

The second stage entails computing Max-Trees [4]. In a Max-Tree, each node represents a connected component $X_h(f)$ of grey-scale image f given by

$$X_h(f) = \{x \in E \mid f(x) \geq h\}, \tag{5}$$

with E the image domain. Because each component of $X_h(f)$ is nested within a larger component $X_{h'}(f)$ with $h' < h$, these components form a tree with the image domain set as the root. It is the equivalent of the *join tree* used in computation geometry [1]. An example is shown in Figure 2. Image (a) shows the original signal, and three sets of peak components (colour coded) that correspond to the decomposition scales or *planes* of the opening instance of the DMP vector field. The radii of the structuring elements used to generate the three planes are 2, 3, and 5. Image (b) shows the Max-Tree of the input signal.

Reconstruction from markers can be computed by building such a Max-Tree, and computing the maximum value of the marker within the image area corresponding to each node. Once the tree has been built, each node receives the maximum marker level within its bounds as output value, unless it is larger than the original value, in which case the node's grey level is unchanged. The parallel algorithm builds a forest of K Max-Trees, one for each disjoint section of the



(a) (b)

Fig. 2. A 1-D signal f (a) partitioned to 3 colour-coded planes of the opening instance of the DMP vector field, and the corresponding Max-Tree (b)

image or volume, and combining the partial Max-Trees into a single one, whilst maintaining the correct marker information [8].

In our implementation, we store the input image in an array f, and the marker image, which will be double as the output image in an array *out*. As in [8], we store the tree in an array of nodes *node* of the same size as the image. Thus each pixel is considered a node in a tree. The parent field of each node contains the index of each pixel's parent. A separate index $\perp$ is used to flag the fact that the node is the root of the tree. Only those nodes n in the final tree, for which the grey value $f[n]$ differs from the parent grey value $f[node[n].parent]$ are considered valid Max-Tree nodes. These nodes, and the root node of the tree are referred to as *level roots* or *canonical elements* of the components represented in the tree. These level roots can be accessed using a function $levroot(n)$ which returns the level root of n, and $Par(n)$ which returns the level root of $node[n].parent$. Each node does require a boolean field *valid*, to indicate if it has already been reconstructed, for the reconstruction phase of the algorithm.

The building phase of the algorithm is essentially the same as that of attribute filters in [8]. The attribute management during the flooding stage of the algorithm is modified, as shown in Alg. 1. Instead of maintaining a single attribute, each node contains a vector of levels to which the node needs to be reconstructed. In Alg. 1, `set` denotes the set of pixels on the boundary of the growing tree. This is stored in a hierarchical queue. We can save memory by using the *out* array to store this vector of reconstruction levels within a node. After the flooding stage, the K trees are merged hierarchically as in [8], again with adaptation to the attribute management, in particular in the *connect* procedure, required to correct a single edge between two adjacent image sections. This adapted algorithm is shown in Alg. 2.

The main difference lies in the filtering phase of the algorithm shown in Alg. 3. For each node within our section we descend the tree, moving from level root to level root, until a node w is found for which either (i) $node[w].valid$ is true, (ii) $Par(w)$ has a marker grey value no smaller than its original; or (iii) $Par(w) = \perp$. In the first case w has been reconstructed previously, so its output value is

Algorithm 1. The modified flooding algorithm for Max-tree construction. Parameter *Vp* indicates the section of the image owned by the current processor *p*, *numscales* indicates the number of scales in the DMP, *out* contains the marker images at input. At output, *out* contains the the correct reconstruction levels at each scale in the canonical elements of each node.

```
procedure flood ($Vp : Section$ ; var $lev : Level$ ; var $thismax : Level[\,]$ ; $f : Image$ ;
                 var $node : Max\text{-}Tree$ ; ; var $out : DMP$)
  for all scales $s$ do
    maximum $[s] := \mathtt{min}(lev, thismax[s])$;
  end ;
  while $\mathtt{set}[lev] \neq \emptyset$ do
    extract some $p$ from $\mathtt{set}[lev]$ ;
    for all scales $s$ do
      maximum $[s] := \mathtt{min}(lev, \mathtt{max}(\mathtt{maximum}[s], out[s][p]))$;
    end ;
    for all neighbours $q \in Vp$ of $p$ do
      if $q \notin \mathtt{set}[fq]$ then
        $fq := f(q)$ ;
        for all scales $s$ do
          childmax $[s] := 0$;
        end ;
        if $\mathtt{lero}[fq] = \perp$ then
          $\mathtt{lero}[fq] := q$ ;
        else $node[q].parent := \mathtt{lero}[fq]$ end ;
        $\mathtt{set}[fq] := \mathtt{set}[f(q)] \cup \{q\}$ ;
        while $fq > lev$ do
          $flood(Vp, fq, childmax, f, node, out)$ end ;
        for all scales $s$ do
          maximum $[s] := \mathtt{min}(lev, \mathtt{max}(\mathtt{maximum}[s], \mathtt{childmax}[s]))$;
        end ;
      end
    end
  end ;
  determine $m$ maximal with $m < lev \wedge (\mathtt{lero}[m] \neq \perp \vee m = -\infty)$ ;
  $node[\mathtt{lero}[lev]].parent := \mathtt{lero}[m]$ ;
  for all scales $s$ do
    $thismax\ [s] := \mathtt{maximum}[s]$;
    $out[s][\mathtt{lero}[lev]] := \mathtt{maximum}[s]$;
  end ;
  $\mathtt{lero}[lev] := \perp$ ;
  $lev := m$ ;
end ;

$\mathtt{hm} := f(\mathtt{xm})$ ; $\mathtt{set}[\mathtt{hm}] := \{\mathtt{xm}\}$ ; $\mathtt{lero}[\mathtt{hm}] := \mathtt{xm}$ ;
for all scales $s$ do
  $\mathtt{maximum}[s] := 0$;
end;
$flood(\mathtt{Vp}, \mathtt{hm}, \mathtt{maximum}, f, node, out)$ .
```

Algorithm 2. The modified connect algorithm.

```
procedure connect($x, y$ : Pixel ; $f$ : Image ; var node : MaxTree)
    for all scales $s$ do maximum[$s$] := 0 ;end;
    $x := levroot(x)$ ;    $y := levroot(y)$ ;
    if $f(y) > f(x)$ then $swap(x, y)$ end
    while $x \neq y \;\wedge\; y \neq \perp$ do
        $z := Par(x)$ ;
        if $z \neq \perp \;\wedge\; f(z) \geq f(y)$ then
            for all scales $s$ do
                $out[s][x] := \min(f[x], \max(out[s][x], \texttt{maximum}[s]))$ ; end;
            $x := z$ ;
        else
            for all scales $s$ do
                temp := $\min(f[x], \max(out[s][x], \texttt{maximum}[s]))$ ;
                maximum[$s$] := $\min(f[x], out[s][x])$ ; $out[s][x]$ := temp ;
            end;
            par[$x$] := $y$; $x := y$ ;    $y := z$ ;
        end
    end
    if $y = \perp$ then
        while $x \neq \perp$ do
            for all scales $s$ do
                $out[s][x] := \min(f[x], \max(out[s][x], \texttt{maximum}[s]))$ ; end;
            $x := Par(x)$ ;
        end
    end
end .
```

correct, in the second case w is reconstructed to the maximum of $f[Par(w)]$ and $out[w]$, unless this is larger than $f[w]$, in which case $f[w]$ becomes the reconstructed value. In the final case the marker image was empty, and zero is chosen as the output value. In all cases the reconstruction value of w is propagated up the tree.

We can use this adaptation of [8] to reconstruction from markers to compute the DMP vector field by direct implementation of (2). However, this entails repeated building of the same Max-Tree. Instead, we can maintain information on multiple markers simultaneously in the Max-Tree, as shown in Alg. 1. Note that as in the case of the original algorithm in [8], the current adaptation works in any number of dimensions, simply by changing the neighbour relationship. The complete algorithm can be summarized as follows:

- Compute all markers using parallel erosions and store each scale i in an image $out[i]$;
- Compute a single Max-Tree in parallel, maintaining the maximum marker for all scales i for each node n in $out[i][n]$;

Algorithm 3. Concurrent implementation of the reconstruction phase. Image *out* contains the maximum marker value within each level root of Max-Tree *node* at input. At output, all *valid* fields in section Vp are set to true, and section Vp of image *out* contains the reconstruction.

```
procedure MaxTreeReconstruct ($Vp$ : Section; var node : Max-Tree; var out : Image)
    for all $v \in Vp$ do
        if not $node[v].valid$ then
            $w := v$;
            while $Par(w) \neq \bot \wedge$ not $node[w].valid \wedge f[Par(w)] > out[Par(w)]$ do
                $w := Par(w)$;
            end;
            if $node[w].valid$ then
                $val := out[w]$;  (* Reconstructed node found *)
            else if  $Par(w) \neq \bot$ then
                    $val := out[w] \vee f[Par(w)]$;   (* $w$ is reconstructed *)
                else  (* marker was empty *)
                    $val := 0$;
                end;
            end;
            for all $u$ in root path from $v$ to $w$ inclusive do
                if $u \in Vp$ then
                    $out[u] := val$;
                    $node[u].valid :=$ true;
                end;
            end;
        end;
    end;
end.
```

- For all scales i
  - Set all nodes to invalid indicating that they have not been filtered yet (in parallel);
  - Compute reconstruction (in parallel) for scale i, storing the result in $out[i]$;
- Compute differences between scales (in parallel);
- Output respective instance of DMP vector field.

Between the different stages a barrier is needed for synchronization of the threads, and within the loop over the scales in the reconstruction phase, a barrier is needed at each iteration. The multithreaded algorithm was implemented using mutexes from the POSIX pthread library. The closing instance is computed by inverting the image and repeating the first two stages.

The marker phase has a computational complexity of $O(R_{\max}N/N_p)$ with $N = X \times Y$ the number of pixels, N_p the number of processors, and $R_{\max}$ the maximum radius of the structuring elements. Complexity in terms of memory

Algorithm 4. The improved reconstruction phase, with same parameters as before, except for *out*. This now contains the maximum marker values for all scales in the DMP within each level root of Max-Tree *node* at input. At output, section *Vp* of DMP *out* contains the morphological profile.

```
procedure $MaxTreeReconstruct$ ($Vp$ : $Section$; var $node$ : $Max\text{-}Tree$; var $out$ : $DMP$)
    for all $v \in Vp$ do
        if not $node[v].valid$ then
            $w := v$;
            for all scales $i$ increasing order do
                while $Par(w) \neq \perp \wedge$ not $node[w].valid \wedge f[Par(w)] > out[Par(w)]$ do
                    $w := Par(w)$;
                end;
                $ws[i] := w$; (∗ temporary storage of for each scale ∗)
                if $node[w].valid$ then
                    for all scales $j \geq i$ do
                        $val[j] := out[j][w]$; (∗ Reconstructed node found ∗)
                        $ws[j] := w$;
                    end;
                else if $Par(w) \neq \perp$ then
                        $val[i] := out[i][w] \vee f[Par(w)]$; (∗ $w$ is reconstructed ∗)
                    else (∗ marker was empty ∗)
                        for all scales $j \geq i$ do
                            $val[j] := 0$;
                            $ws[j] := w$;
                        end;
                    end;
                end;
            end;
            $u := v$;
            for all scales $i$ increasing order do
                repeat
                    if $u \in Vp$ then
                        for all scales $j < i$ do $out[j][u] := f[u]$;end;
                        for all scales $j \geq i$ do $out[j][u] := val[j]$; end;
                        $node[u].valid :=$ true;
                    end;
                    $u := node[u].parent$;
                until $u = ws[i]$;
            end;
            if $u \in Vp$ then (∗ Process $ws[numscales - 1]$ ∗)
                for all scales $j < i$ do $out[j][u] := f[u]$;end;
                for all scales $j \geq i$ do $out[j][u] := val[j]$; end;
                $node[u].valid :=$ true;
            end;
        end;
    end;
end.
```

is $O(N_p R_{\max}^2 X + N)$ [5]. The computational complexity of the local Max-Tree building phase is $O(N(C+G)/N_p)$, with G the number of grey levels, and C the connectivity (4, or 8) [8]. The merging phase has a worst-case time complexity $O(XG \log N \log N_p)$, and filtering has a complexity of $O(N_s N/N_p)$, with N_s the numer of scales [8]. Memory complexity of the Max-Tree part of the algorithm is just $O(N + N_p G)$.

One concern in the above algorithm is the large number of barriers (one per scale). Therefore, instead of performing a single reconstruction using Alg. 3, we produced a second parallel algorithm which produces all reconstructions simultaneously. In this case, only two barriers are needed: one between the marker computation and reconstruction phases, and one between the reconstruction and scale-differencing. We pay for this by a more complex reconstruction algorithm, as shown in Alg. 4. However, since this only traverses the Max-tree once, further speed gains even on a single thread are expected.

The basis of the algorithm is the same, but we now descend the tree until a reconstructed pixel in the largest scale (which means lowest grey-level) marker has been reached. We need to maintain an array *ws* which holds the corresponding values of w in the previous algorithm for each scale, and likewise for the reconstruction value *val*.

Algorithm 5. Concurrent computation of the DMP, thread p.

```
process parDMP($p$ : ThreadID ; $f$ : Image ; var node : Max-Tree ; var out : DMP)
    $find$ xm $such\ that$ $f(\mathtt{xm}) \leq f(x)\ \forall x \in V^p$;
    hm := $f$(xm) ; set[hm] := {xm} ; lero[hm] := xm ;
    for all scales $s$ do
        maximum[$s$] := 0;
    end;
    flood(Vp, hm, maximum, $f$, node, out);
    var $i := 1$ , $q := p$ ;
    while $p + i < K \wedge q$ mod $2 = 0$ do
        $P$(sa[$p + i$])   (* wait to glue with right-hand neighbor *) ;
        for all edges $(x, y)$ between $V(p)$ and $V(p + i)$ do
            connect($x, y, f$, node) ;
        end ;
        $i := 2 * i$ ; $q := q/2$ ;
    end ;
    if $p = 0$ then   (* release the waiting threads *)
        for $i := 1$ to $K - 1$ do $V$(sb[$i$]) end
    else
        $V$(sa[$p$])   (* signal left-hand neighbor *) ;
        $P$(sb[$p$])   (* wait for thread 0 *)
    end ;
    MaxTreeReconstruct(Vp, node, out) ;
end.
```

The complete algorithm for a single thread of the DMP is given in Alg. 5. The first stage, up to and including the call to the *flood* function is the Max-tree construction phase. After this, we enter the hierarchical merging stage, followed by a simple call to the reconstruction function. Note that the implementation of *MaxTreeReconstruct* of Alg. 4 is used in Alg. 5. The code for the complete algorithm is available upon request.

4 Experiments

The first algorithm was tested on a very high resolution (VHR) satellite image of Legaspi (Philippines). The image is a panchromatic tile of 8-bit resolution, it consists of 5998 × 5998 elements (approx. 36MB), and it is of 0.6m × 0.6m spatial resolution. The original was acquired on the 7^{th} of November 2005 and is courtesy of the original Quickbird Imagery @ DigitalGlobe, Inc; 2005 Distributed by Eurimage. A resampled version of the original is shown in the tile of Fig. 4(a). The machine used for timing both processes was a 2 Intel Xeon Quad-Core X5470 (3,3GHz, 1.333MHz FSB) with a total of 16GB DDR2 Quad Channel FBD system memory.

Computing a nine scale DMP using nine separate sequential reconstructions takes 250.1 s on this machine, of which 132.3 s entails computation of the markers. We used markers computed from erosions by Euclidean discs of radius 1 to 256, in factor 2 increments. This is slower than more usual squares but guarantees rotation invariance. The proposed algorithm takes 171.4 s on a single thread. Because the gain is purely in the reconstruction phase, this means a threefold increase in speed of reconstruction (118.1 vs 39.4 s).

The timing results are shown in Fig. 4. At 4 threads the computation time drops to 53.48 s (speed up 3.2): 38.92 s for marker computation (speed-up 3.4), and 14.56 s for reconstruction (speed up 2.7). The overall efficiency is 80%. At 8 threads the speed up is less impressive: 5.1 for the markers, 2.8 for reconstruction, 4.3 total. Overall speed gain with respect to separate computation of each reconstruction is 6.26 overall (5.1 for the markers, and 8.5 for the reconstructions). The low speed-up on 8 cores of the reconstruction phase is certainly partly

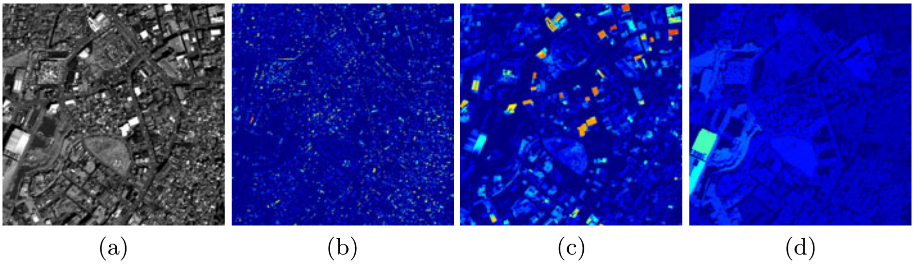


Fig. 3. A panchromatic Quickbird image of the city of Legaspi (a). Three indicative planes of the opening instance of the DMP vector field for $\lambda_i = 5, 25, 60$ for $i = 1, 2, 3$, in (b-d) respectively. λ in this case is the side length of a square structuring element.

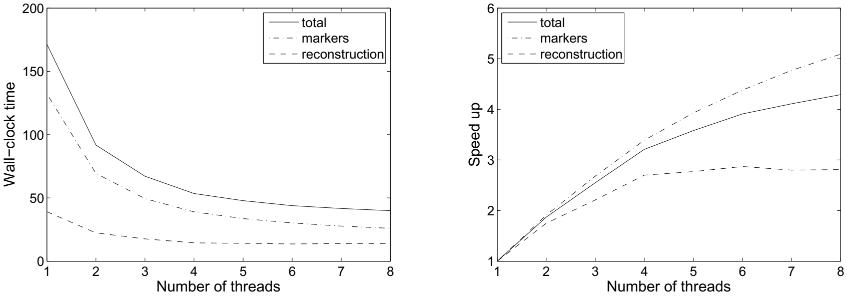


Fig. 4. Wall-clock time (left) and speed up (right) as a function of number of threads, for a 5998 × 5998 image. Timings performed on an 8 core Intel Xeon system.

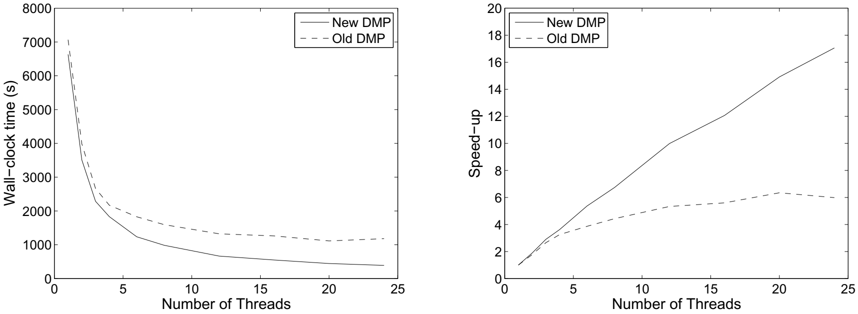


Fig. 5. Timings for the new version of the algorithm with reduced number of barriers: Wall-clock time (left) and speed up (right) as a function of number of threads, for a 1.2 GB image, for the new and old algorithms, for a 24-core AMD Opteron-based machine.

due to the many barriers, but the fact that the trivially concurrent marker computation has an efficiency of 64% suggests that the front-side bus architecture is leading to a memory band-width bottleneck.

For this reason, both algorithms were tested a 24-core Opteron-based machine (4×6 cores) with 128 GB of memory on a 1.2 GB image obtained by repeatedly tiling a 1500 × 1500 satellite image of Haiti. The speed-up was determined for 1 to 24 threads, for both algorithms. What is evident from the results in Figure 5 is that reducing the number of barriers in the algorithm increases efficiency dramatically. On a 24-core Opteron-based machine, we reach a speed up of more than 17 on 24 threads, allowing multi-scale analysis of a 1.2 GB data set in just 6 minutes and 29 seconds, compared to a speed up of just under 6, and computing time of 20 minutes, for the older algorithm. The single sweep of the Max-tree used by the second algorithm also provides a modest increase in speed on a single thread, requiring 1h 50m 34s, rather than 1h 57m 47s, or a 6.5% gain.

5 Conclusions

In this paper we present an algorithm for parallel computation of both reconstruction from markers, and the DMP. Speed-up up to 4 threads on the Intel machine was good, beyond 4 the efficiency dropped off rapidly, in part due to the FSB architecture which limits memory bandwidth, because even the marker computation phase shows only 5.1× speed up on 8 cores. In the other phases, synchronization issues played a part, as was shown in the tests on the AMD Opteron machine. Reducing the number of barriers in the algorithm clearly increases efficiency from 6 to 17 on a 24 core machine. The single sweep of the Max-tree required by the latter algorithm reduces the computing time even on a single core. Beyond these improvements, computation of markers might also be improved by using the algorithm of Vaz et al. [6]. This would improve the speed (if not the speed up) of the marker computation phase, which currently takes up some 50 to 75% of the total computation time. As it stands, we obtain significant gains in speed with respect to the original, sequential approach. Compared to repeatedly computing reconstructions, the total speed-up is 25.1 on 24 cores.

References

1. Carr, H., Snoeyink, J., Axen, U.: Computing contour trees in all dimensions. Comput. Geom. 24, 75–94 (2003)
2. Pesaresi, M., Benediktsson, J.A.: Image segmentation based on the derivative of the morphological profile. In: Goutsias, J., Vincent, L., Bloomberg, D. (eds.) Mathematical Morphology and its Applications to Image and Signal Processing. Computational Imaging And Vision, vol. 18, pp. 179–188. Kluwer Academic Publishers, Boston (2000)
3. Pesaresi, M., Benediktsson, J.A.: A new approach for the morphological segmentation of high-resolution satellite imagery. IEEE Trans. Geosci. Remote Sens. 39(2), 309–320 (2001)
4. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Trans. Image Proc. 7(4), 555–570 (1998)
5. Urbach, E.R., Wilkinson, M.H.F.: Efficient 2-D gray-scale morphological transformations with arbitrary flat structuring elements. IEEE Trans. Image Proc. 17, 1–8 (2008)
6. Vaz, M.S., Kiraly, A.P., Mersereau, R.M.: Multi-level decomposition of Euclidean spheres. In: Proc. Int. Symp. Math. Morphology (ISMM) 2007, pp. 461–472 (2007)
7. Vincent, L.: Morphological grayscale reconstruction in image analysis: application and efficient algorithms. IEEE Trans. Image Proc. 2(2), 176–201 (1993)
8. Wilkinson, M.H.F., Gao, H., Hesselink, W.H., Jonker, J.E., Meijster, A.: Concurrent computation of attribute filters using shared memory parallel machines. IEEE Trans. Pattern Anal. Mach. Intell. 30(10), 1800–1813 (2008)
9. Wilkinson, M.H.F., Soille, P., Ouzounis, G.K.: Concurrent computation of differential morphological profiles for remote sensing. In: Köthe, U., Montanvert, A., Soille, P. (eds.) Proc. of WADGMM ICPR Workshop, pp. 67–71. IAPR, Istanbul (2010)

Hierarchical Segmentation of Multiresolution Remote Sensing Images

Camille Kurtz[1], Nicolas Passat[1], Anne Puissant[2], and Pierre Gançarski[1]

[1] Université de Strasbourg, LSIIT, UMR CNRS 7005, Strasbourg, France
[2] Université de Strasbourg, LIVE, ERL CNRS 7230, Strasbourg, France
{ckurtz,passat,gancarski}@unistra.fr, anne.puissant@live-cnrs.unistra.fr

Abstract. The extraction of urban patterns from very high spatial resolution optical images presents challenges related to the size, the accuracy and the complexity of the data. In order to efficiently carry out this task, a multiresolution hierarchical approach is proposed. It enables to progressively segment several images (of increasing resolutions) of a same scene, based on low level criteria. The process, based on binary partition trees, is partially performed in an interactive fashion, and then automatically completed. Experiments on urban images datasets provide encouraging results which may be further used for detection and classification purpose.

Keywords: Hierarchical segmentation, multisource images, multiresolution, interactive/automated segmentation, partition-trees, remote sensing, urban analysis.

1 Introduction

A new generation of sensors of submetric resolution has led to the production of Very High Spatial Resolution (VHSR) images, and to an improved ability to analyse urban scenes [12]. In particular, in such images, basic urban patterns (*e.g.*, individual houses, gardens, roads) are formed by different materials, while complex ones (*e.g.*, urban districts, urban blocks) generally contain different kinds of basic patterns. Thus, all of them are not necessarily composed of homogeneous pixels (but are often hierarchically organised). These specific properties of VHSR images lead to new challenges for human experts (since the size and the complexity of the images make visual analysis a time consuming and error prone task), and for image analysis tools (since those developed for lower resolutions are generally designed to extract segments based on radiometric homogeneous hypotheses).

In this context, and due to the importance to analyse VHSR images, it is then useful to develop tools adapted to the extraction of complex patterns from such data, and in particular (low-level) segmentation ones. Moreover, the availability of data with a large range of spatial resolutions can enable the extraction of potentially hierarchical patterns, especially when such data are provided by different acquisition devices, providing complementary information at distinct

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 343–354, 2011.

radiometric bands. Such segmentation tools should allow the end-user to obtain satisfactory results, at different levels (*i.e.*, scales) of pattern extraction, with minimal time (by automating the tasks which do not require human expertise), minimal efforts (by reducing the parameters), and ergonomic interaction.

We propose to take advantage of the data available at several resolutions, and to involve them in a hierarchical strategy which enables, at any given resolution, the exploration of the whole structure of an urban scene. This approach, based on the skills of the end-user, aims at using hierarchical segmentation (and especially partition trees) to make the segmentation process as automated as possible.

The article, which is an improved version of the preliminary work described in [8], is organised as follows. Section 2 provides a (non-exhaustive) state of the art on hierarchical and multiresolution segmentation. Section 3 describes the proposed segmentation method. Section 4 gathers experiments enabling to assess the relevance of the approach. Conclusions and perspectives will be found in Section 5.

2 Related Works

Efforts have been conducted to automatically extract features from satellite images, in order to involve them into learning systems. This extraction, often performed thanks to low-level processing, generally relies on radiometric homogeneousness hypotheses. This can lead to valid results for basic objects extraction from High Spatial Resolution (HSR) images [2], but not for images (*e.g.*, VHSR ones) and/or objects of higher complexity [4]. A first way to extract complex objects is by grouping several basic ones, using, for instance, a graph-based approach [3]. Such techniques, devoted to the first semantic level of complex objects (*e.g.*, complex buildings) can be improved by considering hierarchical strategies.

Hierarchical segmentation methods provide, as output, a series of partitions of an image with an increasing (or decreasing) level of details. In the field of remote sensing (and especially for HSR images), several techniques have been proposed. In [11], compositions of opening and closing operations with structuring elements of increasing sizes generate morphological profiles for any pixel, enabling their characterisation. In [1], morphological profiles are enriched with neighbourhood and spectral information. Another method, relying on region connection calculus, can also be found in [6]. These approaches emphasise the potential of hierarchical segmentation. However, these “pixel-based” methods hardly take into account the intrinsic and semantic information of the images. By opposition, “object-based” segmentation hierarchies provide several segmentations of the same image at different levels of detail. Such hierarchies can be built by following two opposite paths. In the top-down approaches, the process starts from a coarse segmentation and successively refines the regions, as in [16], where segmentation is treated as a graph partitioning problem. In the, more frequent, bottom-up approaches, the finest segmentations are produced first, and their regions are then merged, based on similarity criteria [10].

In mathematical morphology, connected operators [15] may be used in a hierarchical segmentation fashion by using, for instance, tree data structures. In such structures, the nodes are associated with regions in the image whereas the edges represent the inclusion relation. Notions such as component-tree [14] and level-lines tree [9], potentially enable to perform hierarchical segmentation, by enabling the fusion of flat zones. However, such structures strongly relying on the image intensity and in particular on extremal values, the obtained segmented components may be non relevant in the case of satellite images. By opposition, the Binary Partition Tree (BPT) [13] reflects a (chosen) similarity measure between neighbouring regions, and models the hierarchy between these regions *via* the tree structure. It has been used to extract complex objects from various kinds of images [21,20]. A last approach, based on the constrained connectivity paradigm, has been recently introduced in [17] and applied to process (V)HSR images in [18]. The connectivity relation generates a partition of the image definition domain. Fine to coarse partition hierarchies are then produced by varying a threshold value associated with each connectivity constraint. In the case of remote sensing, these methods are limited by the spatial and spectral properties of the images. Indeed, complex objects appear in (V)HSR images too much heterogeneous to be extracted in an ascendant way.

This justifies the use of multiresolution data to enhance the behaviour of hierarchical ones for the extraction of such complex objects. Multiresolution methods take advantage of the data available at several resolutions (from Medium Spatial Resolution (MSR) to VHSR ones) [5], and involve them in a hierarchical strategy. By analysing first the image content at a coarse resolution and then gradually increasing this resolution, it is indeed possible to detect complex patterns while avoiding the semantic noise induced by the details [19].

From these considerations, we propose a hierarchical segmentation method, extending the BPT to deal with multiresolution images. It combines the advantages of multiresolution strategies and the efficiency of the connected operators approaches, in the context of the mapping of urban areas. It is based on interactive tree-cut segmentation (based on the skills of the end-user), automatically reproduced on the remainder of the data. The method operates first on the low-resolution data, extracting the global structure of the scene, and subsequently enriches this description thanks to the high-resolution data. It aims, in particular, at understanding the scene in the same way as the human vision system.

3 Methodology

The proposed multiresolution approach (Section 3.2, Figure 2) combines two methods: (1) a new (interactive) hierarchical segmentation process (Section 3.1, Figure 1), and (2) a multiresolution clustering method [7] (Section 3.2.2, Figure 3), into an original iterative process. It is devoted to hierarchically segment n images of a same scene, at various resolutions, from the lowest to the highest one.

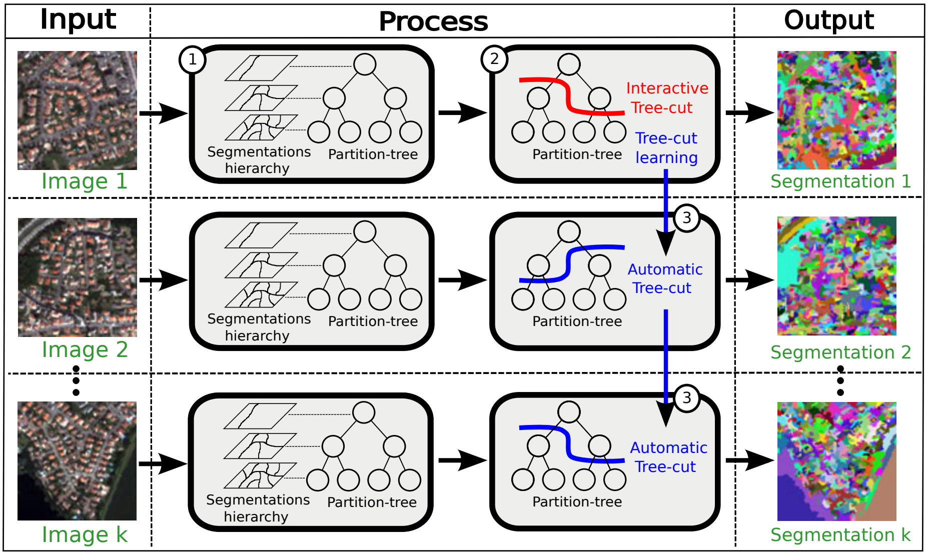


Fig. 1. Interactive segmentation approach (see Section 3.1). In green: input/output. In red: user interactions. In blue: automatic processing.

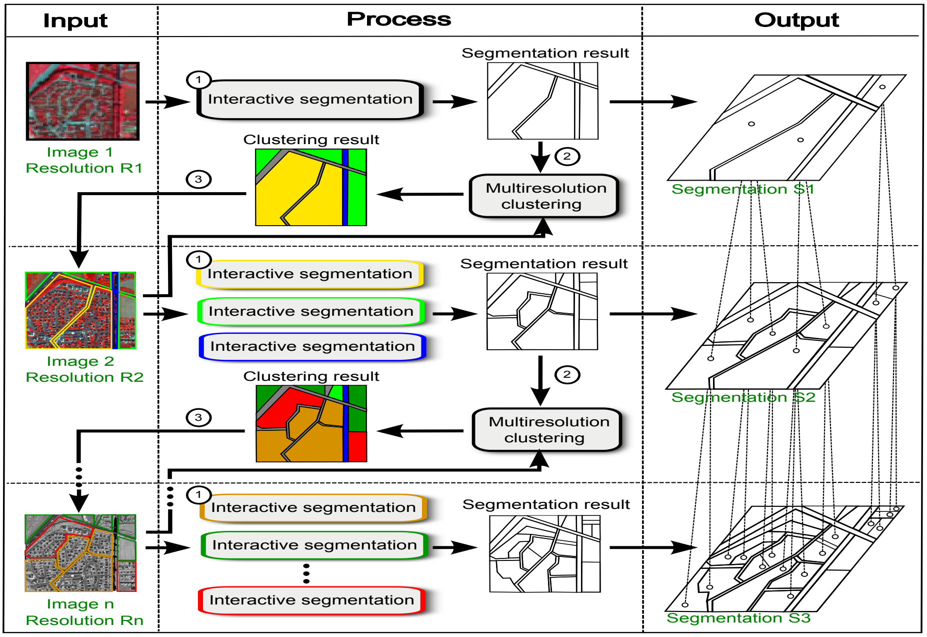


Fig. 2. Work-flow overview (see Section 3.2). In green: input/output.

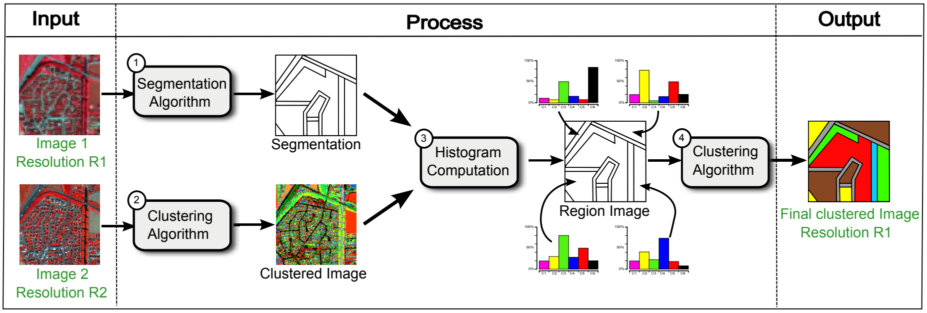


Fig. 3. Multiresolution clustering approach (see Section 3.2.2). In green: input/output.

3.1 Interactive Segmentation

The first contribution is an interactive segmentation approach, enabling to segment k images of same resolution and semantics, provided by the same sensor. From the BPT of one image (Figure 1-①) and an interactive tree-cut of this BPT, inducing a segmentation (Figure 1-②), the $k-1$ other images are automatically segmented, by reproducing a similar tree-cut in their own BPT (Figure 1-③).

3.1.1 Notations

Let $E = [\![0, d_x - 1]\!] \times [\![0, d_y - 1]\!] \subset \mathbb{N}^2$. Let $V_b = [\![0, v_b - 1]\!] \subset \mathbb{N}$. A (monovalue) image is a function $\mathcal{I}_b : E \to V_b$ which to each point $\mathbf{x} = (x, y) \in E$ of the scene, associates a spectral intensity $\mathcal{I}_b(\mathbf{x}) = v$.

Let $V = \prod_{b=1}^{s} V_b \subset \mathbb{N}^s$ $(s \geq 2)$. A (multivalue) image is a function $\mathcal{I} : E \to V$ which to each point $\mathbf{x} = (x, y) \in E$ associates $\mathcal{I}(\mathbf{x}) = \mathbf{v} = \prod_{b=1}^{s} \mathcal{I}_b(\mathbf{x})$.

A segmentation of an image $\mathcal{I} : E \to V$ is a partition $\mathfrak{S} = \{R_i\}_{i=1}^{n}$ $(n \geq 2)$ of E.

3.1.2 Binary Partition Tree

The Binary Partition Tree (BPT) [13] is a hierarchy of regions created by a merging algorithm that can make use of any similarity measure. Starting from a given partition, this algorithm proceeds by (1) computing a similarity measure for all pairs of neighbour regions, (2) merging the most similar pair of regions, and (3) updating the similarity measures (iterating (2,3) until all regions are merged into a single one). The BPT generation then relies on two main notions: the *region model* (which specifies how regions are characterised), and the *merging criterion* (which defines the similarity of neighbouring regions and, thus, the merging order).

Region model. A region R_i is modelled here by (1) $M_{R_i}^{Rad} = \langle (v_{b_{min}}^{R_i}, v_{b_{max}}^{R_i}) \rangle_{b=1}^{s}$, where $v_{b_\star}^{R_i}$ are the extremal radiometric values of the b-th radiometric band of $\mathcal{I}$ (*i.e.*, $\mathcal{I}_b$), and (2) $M_{R_i}^{Geo} = (e(R_i), a(R_i))$, where $e(R_i)$ and $a(R_i)$ are the elongation and the area of (R_i), respectively. During the merging process, the region model of two merged regions R_i, R_j is then straightforwardly provided by

$M^{Rad}_{R_i \cup R_j} = \langle(\min\{v^{R_i}_{b_{min}}, v^{R_j}_{b_{min}}\}, \max\{v^{R_i}_{b_{max}}, v^{R_j}_{b_{max}}\})\rangle^s_{b=1}$ and $M^{Geo}_{R_i \cup R_j} = (e(R_i \cup R_j), a(R_i) + a(R_j))$.

Merging criterion. The basic merging criterion used in most of image segmentation approaches is radiometric homogeneity. Here, we propose to rely on both the increase of the ranges of the intensity values (for each radiometric band) and on area and elongation of the regions in order to merge in priority objects which do not structure the scene:

$$O^{Rad}(R_i, R_j) = \frac{1}{s}\sum_{b=1}^{s} |\max(v^{R_i}_{b_{max}}, v^{R_j}_{b_{max}}) - \min(v^{R_i}_{b_{min}}, v^{R_j}_{b_{min}})| \tag{1}$$

$$O^{Geo}(R_i, R_j) = \frac{1}{2}(e(R_i \cup R_j) + a(R_i \cup R_j)) \tag{2}$$

The similarity measure between two neighbouring regions R_i and R_j is then

$$O(R_i, R_j) = \alpha . O^{Rad}(R_i, R_j) + (1 - \alpha) . O^{Geo}(R_i, R_j) \text{ with } \alpha \in [0, 1] \tag{3}$$

In practice, the closest the nodes are to the root, the less relevant O^{Rad} is. Consequently, the weight α can be defined as a function depending directly on the value of O^{Rad} (and decreasing when O^{Rad} increases). In particular, we have experimentally set $\alpha = \exp(-(O^{Rad})^2)$.

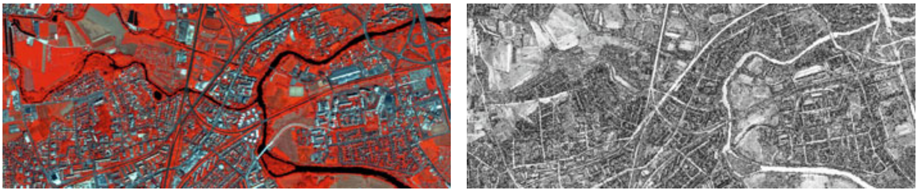
Elongation map. The proposed elongation map characterises the linear structures (roads, rivers, railways, etc.) which generally divide an urban scene into (large) regions. Our purpose, here, is not to get the best elongation results, but to be able to compute correct elongations with a low computational cost. This map is computed as follows:

(1) for each pixel (considered as a seed), a series of region-growing segmentations (based on radiometric intensity) is performed with an increasing tolerance;
(2) for each segmentation, a score is computed using the ratio width/length of the best bounding box of the region (computed in several orientations);
(3) for each pixel, the best (*i.e.*, the lowest) elongation value is assigned.

This approach presents an algorithmic cost bounded, for each pixel, by the area of the neighbourhood where Step (1) is carried out (which, in practice, needs not to be high). The computation of the elongation map is then globally linear with respect to the size of E. Figure 4 provides an example of an elongation map computed on a HSR image with a spatial resolution of 2.4 m and obtained thanks to this heuristic strategy.

3.1.3 Automatic Tree-Cut Learning

Once the BPT built, a cut has to be chosen through the hierarchy. This can be done by performing a thresholding on the similarity measure (called *energy* in the sequel) related to the O function in the BPT (Equation (3)). Broadly speaking, to each node of the tree an energy is attached (by saving for a node R_i, the



(a) HSR image (b) Elongation map

Fig. 4. Elongation map computation. (a) HSR image with a spatial resolution of 2.4 m (QuickBird, © DigitalGlobe Inc.). (b) Corresponding elongation map (elongated structures in light grey, non-elongated ones in dark grey).

value of O required to create this node). A cut can then be easily extracted by determining an adequate energy in a threshold-like fashion, thus preserving the subtree formed by the nodes of higher value (with a possible refinement of one or several branches). This first (manual) step is performed on *one input image* among the k available ones (in red in Figure 1). It is then possible to automatically reproduce this tree-cut operation to segment the $k-1$ other images. This is done by performing a similar tree-cut (*i.e.*, with a same energy) in each one of the $k-1$ corresponding BPTs (in blue in Figure 1).

Without loss of generality, this approach can be applied on k images defined on connected subsets of $\mathbb{N}^2$ (and not only rectangular ones). As a consequence, it enables to segment several sub-parts of a same image. In particular, this is the way it is used in the multiresolution approach described hereafter.

3.2 Multiresolution Strategy

The proposed multiresolution strategy is dedicated to hierarchically segment n images of a same scene at various resolutions, from the lowest to the highest one. In the classical case, three images are considered, namely a MSR (30–5m), a HSR (3–1m) and a VHSR (less than 1m) image. The strategy (Section 3.2.1) combines iteratively the previously described interactive segmentation approach (Section 3.1) and a multiresolution clustering method [7] (Section 3.2.2). At each resolution/step, the output of this process (a segmentation map) is embedded in the next resolution image (using a correspondence map function) to be treated as input of the next step. This process is iterated n times (one per image, from low to high resolution). It provides as final output n segmentations (one per considered image/resolution), hierarchically linked, enabling different scales of interpretation.

3.2.1 Partitioning Strategy at a Given Resolution

At each resolution, the following partitioning strategy is applied. It takes as input one (or more) family(ies) of subimages (with the same semantics). (For instance, in Figure 2, the input used to process the image 2, is composed of three semantic families: the yellow family (composed of two regions of urban areas), the green family (composed of four regions of urban vegetation), and the blue one (composed of one region of water).) First, each family is processed, individually, by the interactive segmentation method (Figure 2-①) generating a segmentation result. (For each family, the user has only to provide one example of tree-cut for a selected region (Section 3.1.3)). These segmentation results are then combined to produce a whole segmentation map. Then, regions provided by this segmentation are gathered into different clusters using a multiresolution clustering method (Figure 2-② and Section 3.2.2) taking as input these regions and the next resolution image. Finally, these classified segments are embedded in the next resolution (Figure 2-③), forming, for each resulting class, a new family of subimages which can be processed by following the same strategy.

3.2.2 Multiresolution Clustering [7]

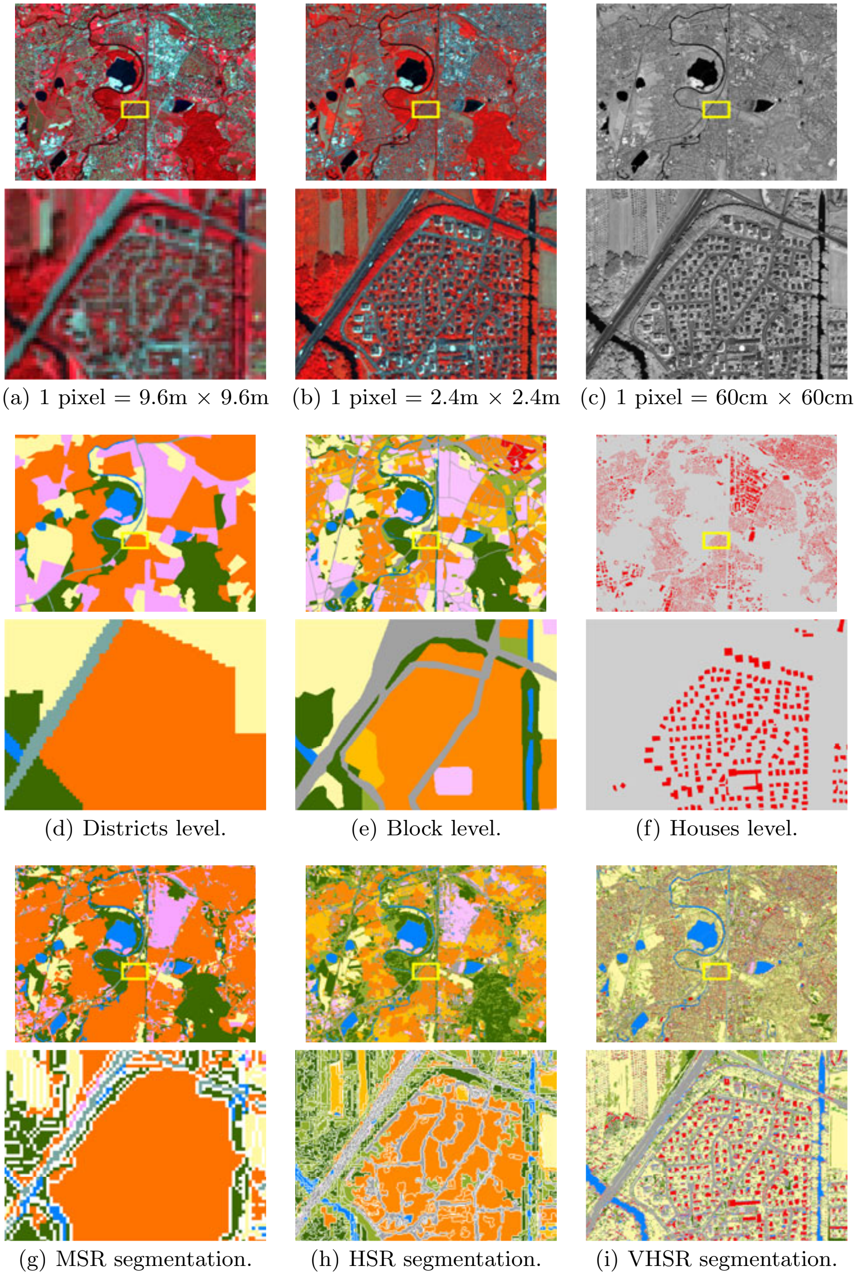
This approach takes as input two multivalue images ($\mathcal{I}^1 : E^1 \rightarrow V^1$ and $\mathcal{I}^2 : E^2 \rightarrow V^2$) of the same scene. The main idea is to fuse the information provided by (1) the analysis of the lowest resolution regions of $\mathcal{I}^1$ (obtained by a segmentation, Figure 3-①) and (2) the highest resolution semantic clustering of $\mathcal{I}^2$ (provided by a classical clustering method directly applied on the radiometric values of the pixels, Figure 3-②), to obtain a final clustering result corresponding to an intermediate level. For each region R_i of the lowest resolution segmented image, an histogram is computed (Figure 3-③) taking into account the distribution of the pixels of R_i in terms of clusters in the highest resolution clustered image. The final clustering result is computed by classifying (in an unsupervised way, Figure 3-④) the regions of the lowest resolution segmented image using their composition histograms. The method finally provides as output a clustering image at an intermediate semantic level (*i.e.*, a level corresponding to a resolution between the ones of $\mathcal{I}^1$ and $\mathcal{I}^2$). This clustering is modelled by a label image $\mathcal{R} : E^1 \rightarrow \llbracket 1, k \rrbracket$ which, to each point $\mathbf{x}$ of the scene (at the lowest resolution), associates a class value $\mathcal{R}(\mathbf{x})$ among the k possible ones. For a more detailed and formalized description of this approach, the reader may refer to [7].

4 Experiments and Results

4.1 Experiments

Experiments have been performed on a multiresolution set of satellite images, presenting a part of the city of Strasbourg, France. These images present a typical suburban environment with water surfaces, forest areas, industrial areas, individual/collective housing blocks and agricultural zones. This set is composed by (1) a single Spot-5 MSR (9.6m) multispectral image (Figure 5(a), 685 × 583 pixels) and by (2) a couple of QuickBird images composed by a HSR (2.4m) multispectral image (Figure 5(b), 2 740 × 2 332 pixels) and a VHSR (60cm) panchromatic one (Figure 5(c), 10 960 × 9 328 pixels).

To assess the efficiency of the proposed approach, several tests have been performed to help end-users to extract a hierarchy of complex urban patterns (urban districts, urban blocks, urban objects) from these data. At each step/resolution,


(a) 1 pixel = 9.6m × 9.6m (b) 1 pixel = 2.4m × 2.4m (c) 1 pixel = 60cm × 60cm

(d) Districts level. (e) Block level. (f) Houses level.

(g) MSR segmentation. (h) HSR segmentation. (i) VHSR segmentation.

Fig. 5. (a–c) Satellite images of the same area (6 500m × 5 400m) with different spatial resolutions: (a) MSR (© CNES–ISIS program), (b) HSR, (c) VHSR (© DigitalGlobe Inc.). (d–f) Corresponding ground-truth maps. (g–i) Obtained segmentations: (g) MSR segmentation, (h) HSR segmentation, (i) VHSR segmentation. A zoom (× 15) on a same given area (the yellow one) is proposed for each image.

the obtained regions have been classified (Figure 5(g–i)) in order to compare them to certified ground-truth maps (Figure 5(d–f)) using the Kappa index.

4.2 Results

Step 1 has been applied on the MSR image to separate the largest structures of the scene (*e.g.*, urban districts, forest areas, water surfaces, etc.). After classification, the comparison between the classified resulting regions (Figure 5(g)) and the ground-truth map has shown a Kappa value of 0.77.

Step 2 has been applied on the HSR image to split these urban districts into different large regions corresponding to: mixed urban districts, commercial or industrial sub-districts, housing blocks, etc. After classification of the resulting regions (Figure 5(h)), the Kappa value for the resulting partition was 0.81.

Step 3 has been performed on the VHSR image to extract "basic" urban objects (*e.g.*, individual/collective houses, vegetations, streets/car parks, shadows, etc.) from these urban blocks. Due to the unavailability of all the "class" information for the ground-truth map corresponding to the VHSR image (only available for the building class, Figure 5(f)), the Kappa index has not been computed. However, after classification of the resulting regions (Figure 5(i)), the percentage of pixels (from the red cluster) matching with the building class was 84% and the percentages of false negatives and false positives were 7% and 16%.

4.3 Runtime

Table 1 provides the runtime and the memory usages for the segmentation of the images. Experiments have been run on an Intel® Core$^{\mathrm{TM}}$2 Quad running at 2.4 GHz with 8 GB of RAM.

As shown by the second column of Table 1, the extraction process is linear with the size of the images. For instance, a HSR image which contains 16 times more pixels than a MSR one, requires 16 times more operations and time to be processed than a MSR one. Since the multiresolution clustering approach (which is mainly based on a partitioning clustering) is linear through the data, we can assume that the interactive segmentation approach is also linear through the data. However, one can see that the memory consumption remains significant when processing the VHSR images (third column of Table 1).

Table 1. Runtime and memory usage for the segmentation of the multiresolution images

Image (Size - pixels)	Extract Runtime	Memory (RAM)
MSR (685 × 583)	23.8 s	54 MB
HSR (2 740 × 2 332)	5 min 49 s	418 MB
VHSR (10 960 × 9 328)	1 h 13 min	2.47 GB

5 Discussion and Perspectives

This article has presented an interactive hierarchical segmentation approach based on binary partition trees and the first results obtained with this methodology on a multiresolution dataset. Experiments have shown that the quality of the extracted urban patterns seems sufficient to further accurately perform both classification or object detection. This seems to validate (1) the relevance of the proposed method and (2) the soundness of the semi-automation of the photo-interpretation approach. However, one can observe that some of the partition results are composed of several regions matching with urban patterns and numerous tiny regions forming linear structures and covering vegetation areas (in particular for the MSR image, Figure 5(g)). These over-segmentation problems are probably due to the spatial criteria used by the algorithm (the elongation one) which does not consider the vegetation areas.

A main advantage of this method is to be parameter-free. Indeed, the only significant parameter is actually the level of tree-cut, the effect of which can be visually assessed by the user. Due to the pre-processing of the data structures, the short computation times enable, in particular, to carry out several segmentations to select the best one.

However, a weakness of this method is that the automation of the interactive tree-cut approach is currently only based on the global energy of the nodes. The next step of this work will then consist of finding a more robust way to reproduce automatically the tree-cut approach in other images. In order to do so, it could be possible to perform nodes matching based on nodes properties and structures.

In a next issue, the results obtained with this method will be fully assessed by quantitative comparisons (using datasets provided by different sensors) and will be compared to the results produced by other hierarchical region-based methods.

References

1. Akcay, H.G., Aksoy, S.: Automatic detection of geospatial objects using multiple hierarchical segmentations. IEEE Transactions on Geoscience and Remote Sensing 46(7), 2097–2111 (2008)
2. Baatz, M., Hoffmann, C., Willhauck, G.: Progressing from object-based to object-oriented image analysis. In: Blaschke, T., Lang, S., Hay, G.J. (eds.) Object-Based Image Analysis. LNCS, ch. 1.2, pp. 29–42. Springer, Heidelberg (2008)
3. Barnsley, M.J., Barr, S.L.: Distinguishing urban land-use categories in fine spatial resolution land-cover data using a graph-based, structural pattern recognition system. Computers, Environment and Urban Systems 21(3), 209–225 (1997)
4. Blaschke, T.: Object based image analysis for remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing 65(1), 2–16 (2010)
5. Gaetano, R., Scarpa, G., Poggi, G.: Hierarchical texture-based segmentation of multiresolution remote-sensing images. IEEE Transactions on Geoscience and Remote Sensing 47(7), 2129–2141 (2009)
6. Inglada, J., Michel, J.: Qualitative spatial reasoning for high-resolution remote sensing image analysis. IEEE Transactions on Geoscience and Remote Sensing 47(2), 599–612 (2009)

7. Kurtz, C., Passat, N., Gançarski, P., Puissant, A.: Multiresolution region-based clustering for urban analysis. International Journal of Remote Sensing 31(22), 5941–5973 (2010)
8. Kurtz, C., Puissant, A., Passat, N., Gançarski, P.: An interactive approach for extraction of urban patterns from multisource images. In: Symposium of JURSE 2011, Joint Urban Remote Sensing Event - 6th Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (to appear, 2011)
9. Monasse, P., Guichard, F.: Scale-space from a level lines tree. Journal of Visual Communication and Image Representation 11(2), 224–236 (2000)
10. Pavlidis, T.: Structural pattern recognition. Springer, Heidelberg (1977)
11. Pesaresi, M., Benediktsson, J.A.: A new approach for the morphological segmentation of high-resolution satellite imagery. IEEE Transactions on Geoscience and Remote Sensing 39(2), 309–320 (2001)
12. Puissant, A., Weber, C.: The utility of Very High Spatial Resolution images to identify urban objects. Geocarto International 17(1), 33–44 (2002)
13. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Transactions on Image Processing 9(4), 561–576 (2000)
14. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. IEEE Transactions on Image Processing 7(4), 555–570 (1998)
15. Serra, J.C., Salembier, P.: Connected operators and pyramids. In: Dougherty, E.R., Gader, P.D., Serra, J.C. (eds.) Image Algebra and Morphological Image Processing IV, vol. 2030, pp. 65–76. SPIE, San Diego (1993)
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
17. Soille, P.: Constrained connectivity for hierarchical image decomposition and simplification. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(7), 1132–1145 (2008)
18. Soille, P.: Constrained connectivity for the processing of very-high-resolution satellite images. International Journal of Remote Sensing 31(22), 5879–5893 (2010)
19. Sun, W., Heidt, V., Gong, P., Xu, G.: Information fusion for rural land-use classification with high-resolution satellite imagery. IEEE Transactions on Geoscience and Remote Sensing 41(4), 883–890 (2003)
20. Valero, S., Salembier, P., Chanussot, J.: New hyperspectral data representation using binary partition tree. In: IEEE International Geoscience and Remote Sensing Symposium, vol. 2, pp. 80–83 (2010)
21. Vilaplana, V., Marques, F., Salembier, P.: Binary partition trees for object detection. IEEE Transactions on Image Processing 17(11), 2201–2216 (2008)

Mathematical Morphology for Vector Images Using Statistical Depth

Santiago Velasco-Forero and Jesus Angulo

CMM-Centre de Morphologie Mathématique
Mathématiques et Systèmes, MINES ParisTech;
35, rue Saint-Honoré, 77305 Fontainebleau CEDEX - France
{velasco,angulo}@cmm.ensmp.fr
http://cmm.ensmp.fr/~velasco;~angulo

Abstract. The open problem of the generalization of mathematical morphology to vector images is handled in this paper using the paradigm of depth functions. Statistical depth functions provide from the "deepest" point a "center-outward ordering" of a multidimensional data distribution and they can be therefore used to construct morphological operators. The fundamental assumption of this data-driven approach is the existence of "background/foreground" image representation. Examples in real color and hyperspectral images illustrate the results.

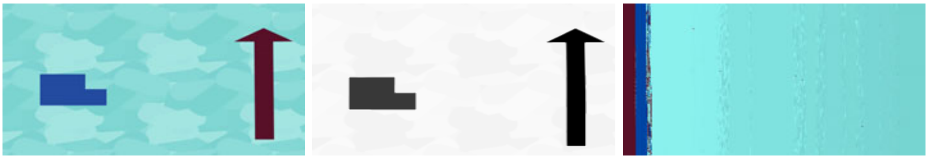
Keywords: Multivariate Morphology, Depth function, Hyperspectral Images.

1 Introduction

Mathematical morphology is a nonlinear image processing methodology based on the application of lattice theory to spatial structures. It requires the definition of a complete lattice structure, i.e., the definition of an ordering among the points to be analyzed. The extension of mathematical morphology to vector spaces, for instance, color/multi/hyper/ultraspectral images, is neither direct nor trivial because the pixels are vectors. An alternative way to approach the problem of multi-channel morphology is to treat the data at each pixel as a vector. Unfortunately, there is no unambiguous means of defining the minimum and maximum values between two vectors of more than one dimension, and thus it is important to define an appropriate arrangement of vectors in the selected vector space. We refer keen readers to [2,3] for a comprehensive review of vector morphology. To overcome the lack of natural ordering, the following four families of ordering for multichannel samples have been identified in the literature [4,13]. The trivial approach consists in applying grayscale morphology techniques to each channel separately, that has been called marginal morphology in the literature [18]. However, the marginal approach is often unacceptable in several applications because, when morphological techniques are applied independently to each image channel, analysis techniques are subject to the well-known problem of false colors [17]; that is, it is very likely that new spectral constituents

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 355–366, 2011.

(not present in the original image) may be created as a result of processing the channels separately. It is called *marginal ordering* (M-ordering). To strictly preserve input vectors, the *conditional ordering* (C-ordering) approach, also known as lexicographic ordering, is frequently used. The C-ordering is based on the ordering of the components selected sequentially according to different conditions or priorities. When all the components are used, the C-ordering is a total ordering. The *reduced ordering* (R-ordering) which performs the ordering of vectors according to some scalars, computed from a mapping of the vector onto a different representation where the ordering is naturally defined, typically distances or projections onto a dimensionality reduced space (using for instance the principal component analysis). For instance, Mahalanobis distance has been employed in several works on multivariate morphology [1,7] but using a reference set. The *P-ordering*, in the original description [4], is based on the partition of the vectors into groups, such that the groups can be distinguished with respect to rank or extremeness. Recently, other approaches using combinatorial techniques and median/anti-median filters have been also used to construct ordering [9,14]. There is however a problem of these latter approaches: the ordering is locally depending on the values of the window, consequently it is not a partial ordering for the set of vector values of the images, i.e., dilation(erosion) obtained does not commute with the supremum(infimum) and the distributive property is not valid. Finally, supervised ordering has shown be useful in the analysis of high dimensional images [20,21]. In this paper, a P-ordering for pixels in multivariate images is presented. To the best of our knowledge, this is the first approach which uses P-ordering to extend mathematical morphology in multivariate image. The basic idea is to produce an ordering by using statistical depth functions. Statistical depth functions provide from the "deepest" point a "center-outward ordering" of multidimensional data. Therefore, the assumption of the existence of background/foreground representation is required. Formally we could express the assumption of *background/foreground representation* in this way: Given a vector image $\mathbf{I} : E \rightarrow \mathbb{R}^d$, the support space E has a decomposition $E = \{E_B, E_F\}$ such that $E_B \cap E_F = \emptyset$ and $\mathbf{card}\{E_B\} > \mathbf{card}\{E_F\}$. Roughly speaking, the assumption says: (1) The image has two main components: the background and the foreground; (2) There are more pixels in the background than in the foreground. We notice that there is not hypothesis about the multivariate or spatial distribution of the background $\mathbf{I}(E_B)$ and the foreground $\mathbf{I}(E_F)$. Following that idea, the aim of this paper is to introduce the depth functions as a clever way to produce a data-driven ordering useful to produce mathematical morphology operators. Fig. 1 shows a toy color image example $\mathbf{I}$ and its depth function associated $D(\cdot; \mathbf{I})$. Thus, the ordering for two pixel vectors is given by $\mathbf{v}_1 < \mathbf{v}_2 \leftrightarrow D(\mathbf{v}_1; \mathbf{I}) > D(\mathbf{v}_2; \mathbf{I})$. The paper is organized as follows. Restricting ourselves to data in vector spaces, in Section 2 the statistical depth functions definition is presented. Section 3 analyzes the application of ordering using depth functions in the context of vector images and it presents some results in color and hyperspectral images. Finally, Section 4 concludes the paper.



(a) Original, $\mathbf{I}$ (b) Depth function of (a), $h(\cdot) = D(\cdot; \mathbf{I})$ (c) Ordering induced by (b)

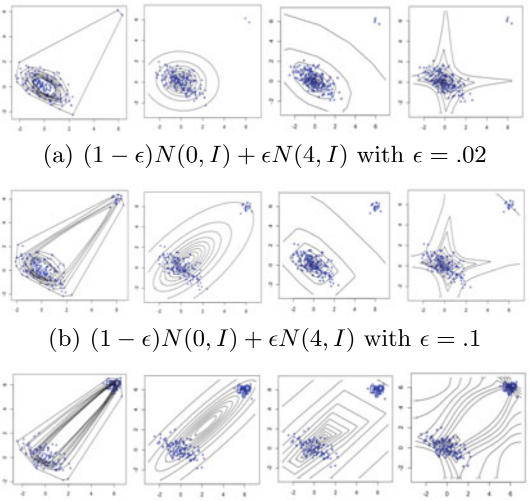
Fig. 1. Intrinsic ordering based on dichotomy background and foreground. $\mathbf{v}_1 < \mathbf{v}_2 \Leftrightarrow D(\mathbf{v}_1; \mathbf{I}) > D(\mathbf{v}_2; \mathbf{I})$

2 Statistical Depth Functions

Depth functions for multivariate data have been pursued in robust and nonparametric data analysis and inference. Depth functions assign to each point its degree of centrality with respect to a data cloud or a probability distribution. A depth function suitable for a distribution $\mathfrak{F}$ in $\mathbb{R}^d$, denoted by $D(\mathbf{v}; \mathfrak{F})$, brings out the non-central ranking of the vector $\mathbf{v}$ in $\mathbb{R}^d$ with respect to $\mathfrak{F}$. A number of depth functions are available in the literature, for instance *halfspace depth* [19], *simplicial depth* [10], *projection depth* [6], *spatial depth* [22], *Mahalanobis depth* [23], etc. Roughly speaking, for a distribution $\mathfrak{F} \in \mathbb{R}^d$, a corresponding depth function $D(\mathbf{v}; \mathfrak{F})$ could provide an $\mathfrak{F}$-based center-outward ordering of point $\mathbf{v} \in \mathbb{R}^d$. Hence, $D(\mathbf{v}; \mathfrak{F})$ is a function $\mathbb{R}^d \to \mathbb{R}$. Depth-based methods are completely data-driven and avoid strong distributional assumption. Moreover, they provide intuitive visualization of the data set via depth contours for a low dimensional input space. Analogous to linear order in one dimension, statistical depth functions provide an ordering of all points from the center outward in a multivariate data set, where the median is the "deepest" point in the data set. This leads to center-outward ordering to points and to a description in terms of nested contours. Let us start by a formal definition of a depth function.

Definition 1. *A statistical depth function is a bounded nonnegative mapping* $D(\cdot; \cdot) : \mathbb{R}^d \times \mathfrak{F} \to \mathbb{R}$ *satisfying*

1. $D(\mathbf{A}\mathbf{v} + \mathbf{b}; \mathfrak{F}_{\mathbf{A}\mathbf{v}+\mathbf{b}}) = D(\mathbf{v}; \mathfrak{F})$ *holds for any random vector* $\mathbf{v}$ *in* $\mathbb{R}^d$*, any* $d \times d$ *nonsingular matrix* $\mathbf{A}$*, and any* $\mathbf{b} \in \mathbb{R}^d$*. That invariance to affine transformation means, the depth of a vector* $\mathbf{v} \in \mathbb{R}^d$ *should not depend on the underlying coordinate system or, in particular, on the scales of the underlying measurements.*
2. $D(\theta; \mathfrak{F}) = \sup_{\mathbf{v} \in \mathbb{R}^d} D(\mathbf{v}; \mathfrak{F})$ *holds for any* $\mathfrak{F}$ *having center* θ*. That means, for any distribution having a unique "center", the depth function should attain maximum value at this center.*
3. $D(\mathbf{v}; \mathfrak{F}) \leq D(\theta + \alpha(\mathbf{v} - \theta); \mathfrak{F})$ *holds for any* $\mathfrak{F}$ *having a deepest point* θ *and any* $\alpha \in [0, 1]$*, i.e., as a point* $\mathbf{v} \in R^d$ *moves away from the "deepest point" along any fixed ray through the center, the depth at* $\mathbf{v}$ *should decrease monotonically.*



(a) $(1-\epsilon)N(0,I)+\epsilon N(4,I)$ with $\epsilon = .02$

(b) $(1-\epsilon)N(0,I)+\epsilon N(4,I)$ with $\epsilon = .1$

(c) $(1-\epsilon)N(0,I)+\epsilon N(4,I)$ with $\epsilon = .4$

Fig. 2. Random sample of a bivariate mixed gaussian distribution with different values of contaminate rate ϵ. Columns correspond with Tukey Depth, Mahalanobis Depth, Projection Depth and Spatial Depth. ϵ is the contaminate data rate.

4. *$D(\mathbf{v};\mathfrak{F}) \to 0$ as $||\mathbf{v}|| \to \infty$, for each $\mathfrak{F}$, i.e., the depth of a point $\mathbf{v}$ should approach to zero as its norm approaches infinity.*

The four precedent properties are introduced and investigated for Liu in [10]. To apply the statistical depth function paradigm in the mathematical morphology context, we define the set of vector pixels $\mathbb{V}$ corresponding to a given vector image $\mathbf{I} : E \to \mathbb{R}^d$ with $n = \mathbf{card}(E)$ pixels as the set $\mathbb{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$, where $\mathbf{v}_i$ is a d-dimensional vector pixel of $\mathbf{I}$. Thus, the function $D(\cdot;\mathbb{V})$ is called a *random depth function* based on $\mathbb{V}$ according with [5]. Additionally, random depth function should be permutation invariant, i.e. $D(\mathbf{v};\mathbb{V}) = D(\mathbf{v};\Pi(\mathbb{V}))$ for all Π a permutation of the elements in $\mathbb{V}$.

2.1 Halfspace Depth

Tukey [19] proposed a "Tukey-depth" and suggested its role in defining multivariate analogues of univariate rank and order statistics via depth-induced "contours".

Definition 2. *The Tukey depth relative to $\mathbb{V}$ of a point $\mathbf{v}$ in $\mathbb{R}^d$ is defined as the smallest number of data points in any closed halfspace with boundary through $\mathbf{v}$. This is also called the halfspace depth, and it can be written as*

$$D_T(\mathbf{v};\mathbb{V}) = \min_{||\mathbf{u}||=1} \mathbf{card}\{i; \mathbf{u}^T\mathbf{v}_i \geq \mathbf{u}^T\mathbf{v}\}$$

where $\mathbf{u}$ ranges over all vector in $\mathbb{R}^d$ with $||\mathbf{u}|| = 1$, and **card** *is the set cardinal.*

In the one-dimensional case, it is reasonable to order the point for a sample $\mathbb{V} = \{v_1, \ldots, v_n\}$ by using the one-dimensional halfspace depth

$$D_T(v; \mathbb{V}) = \min\{\mathbf{card}\{i; v_i \geq v\}, \mathbf{card}\{i; v_i \leq v\}\}. \tag{1}$$

Thus, the points are ordered following the decreasing order of the absolute value of the difference between their percentiles and 50, and the deepest points are the medians of $\mathbb{V}$. Examples of halfspace depth function for some mixed gaussian distribution are shown in Fig. 1. Roughly speaking, the maximum value of the Tukey depth is a median under any linear projection from $\mathbb{R}^d$ to $\mathbb{R}$.

2.2 Spatial Depth

Although both the sample mean and median of a data set are natural estimators for the center of a distribution, the median is insensitive to extreme observations while the mean is highly sensitive. A single contaminating point to a data set can send the sample mean, in the worst case, to infinity, whereas in order to have the same effect on the median, at least 50 percent of the data points must be moved to infinity. Let $s : \mathbb{R} \to \{-1, 0, 1\}$ be the sign function. For $\mathbf{v} \in \mathbb{R}$, the difference between the number of observations on the left and right of $\mathbf{v}$ is given by $|\sum_{i=1}^{n} s(\mathbf{v}_i - \mathbf{v})|$, then the sample median $\mathbf{m}$ satisfies $|\sum_{i=1}^{n} s(\mathbf{v}_i - \mathbf{m})| = 0$. A generalized sign function for multidimensional data is *the spatial sign function*, which is a map $S : \mathbb{R}^d \to \mathbb{R}^d$ given by

$$S(\mathbf{v}) = \begin{cases} \mathbf{v}/||\mathbf{v}||, & \mathbf{v} \neq \mathbf{0}; \\ \mathbf{0}, & \mathbf{v} = \mathbf{0}. \end{cases}$$

where $||\mathbf{v}|| = \sqrt{\mathbf{v}^T\mathbf{v}}$ and $\mathbf{0}$ is the zero vector in $\mathbb{R}^n$. With the spatial sign function, the multidimensional sample median for $\mathbb{V}$ is a straightforward analogy of the univariate version, i.e., it is the sample median $\mathbf{m}$ satisfies $||\sum_{i=1}^{n} S(\mathbf{v}_i - \mathbf{m})|| = 0$. It is called the *spatial median* or the *L_1-median* [22]. The concept of spatial depth was formally introduced by Serfling [15] based on the notion of spatial quantiles, as follows

Definition 3. *For a multivariate cumulative distribution $\mathfrak{F}$ on $\mathbb{R}^n$, the spatial depth of a point $\mathbf{v} \in \mathbb{R}^d$ with respect to the distribution $\mathfrak{F}$ is defined as*

$$D_S(\mathbf{v}; \mathfrak{F}) = 1 - \left|\left| \int S(\mathbf{w} - \mathbf{v}) d\mathfrak{F}(\mathbf{w}) \right|\right|$$

Therefore, given set of vectors $\mathbb{V}$ the *sample spatial depth* is defined as follows

$$D_S(\mathbf{v}; \mathbb{V}) = 1 - \frac{1}{\mathbf{card}(\mathbb{V} \cup \{\mathbf{v}\}) - 1} \left|\left| \sum_{\mathbf{w} \in \mathbb{V}} S(\mathbf{w} - \mathbf{v}) \right|\right| \tag{2}$$

Note that (2) has a range $[0, 1]$. The spatial median is a set of data points that have the "deepest" depth 1. Indeed, the spatial depth provides from the "deepest" point a "center-outward" ordering of multidimensional data. Examples of spatial depth function for some mixed gaussian distribution are illustrated in Fig. 1. Unfortunately, the L_1-median is not affine invariant (Property 1, Definition 1). However, the spatial depth can be affine invariant if a suitable data-dependent coordinate system is used, e.g., to replace $S(\mathbf{w}-\mathbf{v})$ throughout by $S(\mathbf{A}(\mathbf{w}-\mathbf{v}))$ where $\mathbf{A}=\Sigma^{-1/2}$ for a suitable robust covariance-type matrix Σ based on data [22].

2.3 Projection Depth

Introduced by Y. Zuo[23], it defines the outlyingness of vector $\mathbf{v}$ to be the worst case outlyingness with respect to the one-dimensional scale functional in any one-dimensional projection, that is,

Definition 4. *The projection depth of* $\mathbf{v}\in\mathbb{R}^d$ *with respect to* $\mathbb{V}$ *is defined as*

$$D_P(\mathbf{v};\mathbb{V})=\frac{1}{1+\sup_{||\mathbf{u}||=1}\frac{|\mathbf{u}^T\mathbf{v}-\mu(\mathbb{V}_u)|}{\sigma(\mathbb{V}_u)}},$$

where μ *and* σ *are location and scale functionals and* $\mathbb{V}_{\mathbf{u}}$ *denotes the value of* $\mathbf{u}^T\mathbb{V}=\{\mathbf{u}^T\mathbf{v_1},\ldots,\mathbf{u}^T\mathbf{v_n}\}$.

The outlyingness is the largest standardized deviation from the center among all univariate projections. For robustness, μ and σ should be robust estimators, for example, the median and the median absolute deviation (MAD). Projection depth for mixed gaussian distribution are illustrated in Fig. 1. For more properties and details about projection depth, see [23].

2.4 Mahalanobis Depth

Zuo and Serfling [23] proposed the Mahalanobis depth as example of a function with the desirable properties presented in the definition 1.

Definition 5. *The* Mahalanobis depth *of* $\mathbf{v}\in\mathbb{R}^d$ *with respect to* $\mathbb{V}$ *is defined as*

$$D_M(\mathbf{v};\mathbb{V})=\frac{1}{1+d^2_{\Sigma_{\mathbb{V}}}(\mathbf{v},\mu_{\mathbb{V}})},$$

where $d^2_{\Sigma_{\mathbb{V}}}(\mathbf{v},\mu_{\mathbb{V}})=(\mathbf{v}-\mu_{\mathbb{V}})^T\Sigma_{\mathbb{V}}^{-1}(\mathbf{v}-\mu_{\mathbb{V}})$*, and* $\mu_{\mathbb{V}}$ *and* $\Sigma_{\mathbb{V}}$ *are any multivariate location and covariance measures of the set of vectors in* $\mathbb{V}$.

The case with $\mu_{\mathbb{V}}$ and $\Sigma_{\mathbb{V}}$ being the mean and covariance matrix of $\mathbb{V}$ is the well-known Mahalanobis distance. For these choices, $D_M(\mathbf{v};\mathbb{V})$ is not robust, and it can fail to achieve the item 3 in the definition 1 for some family of theoretical distribution, for example, a symmetric distribution. However, the sample version satisfies all of the five desirable properties, as it has shown by Zuo and Serfling [23].

2.5 Practical Implementation

Statistical depth functions have interesting theoretical properties as it has been intensively illustrated in previous subsections. They induce a data-driven ordering avoiding the assumption of some probability distribution. An example is shown in Fig. 2 for the case of mixed gaussian distribution. The lack of robustness for the Mahalanobis depth is an important point to remark. Halfspace, spatial and projection depth have higher robustness, i.e., they are preserving the maximum value in the center of the largest distribution even with a large percentage of contaminate data (ϵ in Fig. 2). Unfortunately, halfspace and projection depth require high computation time for an exact solution. This is more or less reasonable if $d = 2$, but it becomes prohibitive even for $d > 7$ [11]. However, in most cases, the approximate computation by stochastic sampling is fast and accurate enough for general applications. Therefore, we propose to compute the maximum outlyingness among K_1 random directions in $\mathbb{S}^{d-1}$ the d-dimensional unitary ball. We must choose K_1 large enough to obtain stability of the results. These issues are summarized in Table 1. The complexity of spatial depth function is $O(n^2)$, where $n = \mathbf{card}(\mathbb{V})$. However it is relatively slow to calculate in real images. In the practical examples, a random sequential sample of size K_2 denoted by $\hat{\mathbb{V}}$ is selected from $\mathbb{V}$ to calculate the spatial depth function in reasonable time.

Table 1. Approximated version used in the experiments

Depth function	Approximated Version	Parameter
Half-space	$\min_{\mathbf{u}\in\mathbb{S}} \mathbf{card}\{i; \mathbf{u}^T\mathbf{v}_i \geq \mathbf{u}^T\mathbf{v}\}$	$K_1 = \mathbf{card}(\mathbb{S})$
Projection	$\sup_{\mathbf{u}\in\mathbb{S}} \frac{\lvert\mathbf{u}^T\mathbf{v}-\mu(\mathbb{V}_u)\rvert}{\sigma(\mathbb{V}_\mathbf{u})}$	$K_1 = \mathbf{card}(\mathbb{S})$
Spatial	$\sum_{\mathbf{w}\in\hat{\mathbb{V}}} S(\mathbf{w}-\mathbf{v})$	$K_2 = \mathbf{card}(\hat{\mathbb{V}})$
Mahalanobis	Exact	none

3 Morphology Using Statistical Depth Functions

The basic idea of our approach is to use a depth function to induce an h-ordering in the vector space. That is an ordering based on a data-adapted function and in such way that the interpretation of max and min operation in this lattice is

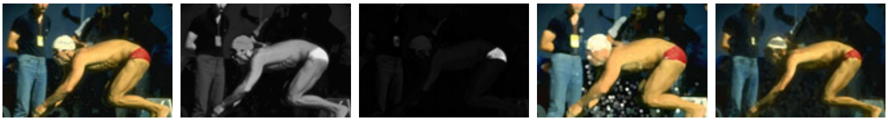
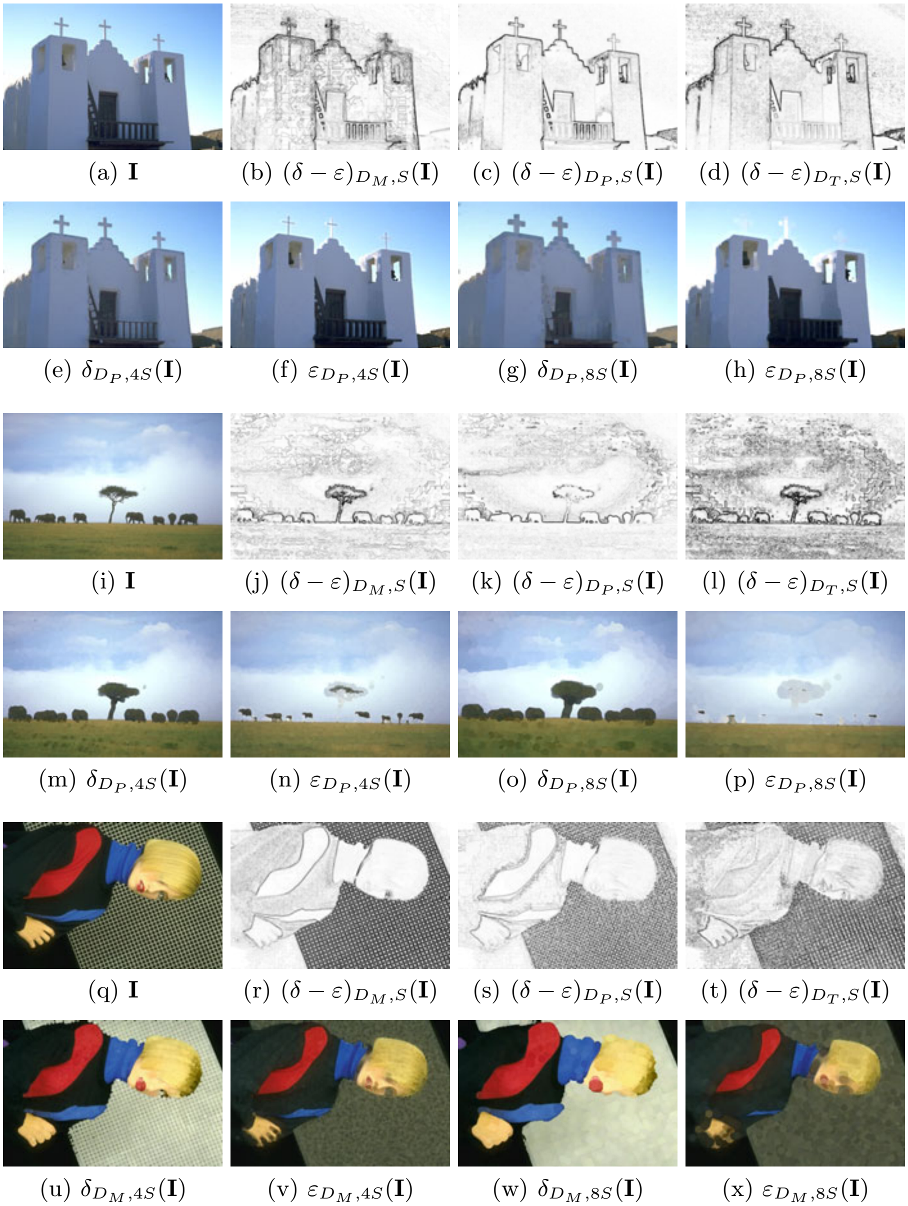

(a) $\mathbf{I}$ (b) $D_P(\cdot;\mathbf{I})$ (c) $D_M(\cdot;\mathbf{I})$ (d) $\delta_{D_P,5S}(\mathbf{I})$ (e) $\varepsilon_{D_P,5S}(\mathbf{I})$

Fig. 3. The lack of robust of depth Mahalanobis D_M whereas projection depth D_P involves an ordering useful for dilations/erosions

known a priori, because max values can be associated with "outlier" pixels in the high-dimensional space and min are "central" pixels in $\mathbb{R}^d$ space. In this section, fundamentals of complete lattices for $\mathbb{R}^d$ is reviewed. For a detailed exposition on complete lattice theory in mathematical morphology refer to [12]. A space $\mathcal{L}$ endowed with a partial order $\leq$ is called a *complete lattice*, denoted $(\mathcal{L}, \leq)$ if every subset $\mathcal{H} \subseteq \mathcal{L}$ has both supremum (join) $\bigvee \mathcal{H}$ and infimum (meet) $\bigwedge \mathcal{H}$. A *minimum (smallest)* $n \in \mathcal{H}$ is an element contained in all other elements of $\mathcal{H}$, that is, $l \in \mathcal{H} \Rightarrow n \leq l$. We denote the minimum of $\mathcal{L}$ by $\perp$. Equivalently, a *maximum (largest)* n in $\mathcal{H}$ is an element that contains every element of $\mathcal{H}$, that is, $l \in \mathcal{H} \Rightarrow l \leq n$. We denote the maximum of $\mathcal{L}$ by $\top$. Let R be a nonempty set and $\mathcal{L}$ a complete lattice. Furthermore, let $h : R \rightarrow \mathcal{L}$ be a surjective mapping. As it was defined in [8], we refer by $\leq_h$ as the *h-ordering* given by:

$$r \leq_h r' \Leftrightarrow h(r) \leq h(r'), \quad \forall r, r' \in R$$

Note that $\leq_h$ preserves reflexivity ($r \leq_h r$) and transitivity ($r_1 \leq_h r_2$ and $r_2 \leq_h r_3 \Rightarrow r_1 \leq_h r_3$) but is not a total ordering. Additionally, an equivalence class is defined by $\mathcal{L}[z] = \{r \in R | h(r) = z\}$. For multivariate images, as color or hyperspectral ones, pixel values are vectors defined in $\mathbb{R}^d$. Consequently the main challenge to build complete lattice structures is to define a mapping $h : \mathbb{R}^d \rightarrow \mathcal{L}$, where $\mathcal{L}$ can be the lattice of the extended real line $(\overline{\mathbb{R}}, \leq)$ with $\overline{\mathbb{R}} = \mathbb{R} \bigcup \{-\infty, +\infty\}$ and $\leq$ as the "less than or equal to" operation (the natural partial ordering). Once the family of orderings have been established, the morphological vector operators are defined in the standard way. We limit here our developments to the flat operators, i.e., the structuring elements are planar shapes. Let $\mathbf{I} : E \rightarrow \mathbb{R}^d$ be a vector image. According to the previous developments, we consider that there exists a mapping $h : \mathbb{R}^d \rightarrow \mathcal{L}$, furthermore the composition of $\mathbf{I}$ and h will be $h(\mathbf{I}) : E \rightarrow \mathcal{L}$. The functions from E onto $\mathcal{L}$ are denoted by $\mathcal{F}(E, \mathcal{L})$, where $E \subset \mathbb{Z}^2$ is the support space of the image. The partial ordering in lattice $\mathcal{L}$ is denoted $\leq_{\mathcal{L}}$. If $\mathcal{L}$ is a complete lattice, then $\mathcal{F}(E, \mathcal{L})$ is a complete lattice too. We refer $h_{D(\cdot, \mathbf{I})}$ as the ordering generated for a depth function $-D(\cdot, \mathbf{I})$ (By a mild abuse of notation we write $-D(\cdot; \mathbf{I})$, but naturally, the depth function is computed in the corresponding set $\mathbb{V}$ as it was presented in Section 2). Morphological operators defined using $h_{D(\cdot, \mathbf{I})}$ have the classical properties [18]. Under the assumption that the original image $\mathbf{I} : E \rightarrow \mathbb{R}^d$ has *background/foreground decomposition*, i.e., $E = \{E_B, E_F\}$ such that $E_B \cap E_F = \emptyset$ and $\mathbf{card}\{E_B\} > \mathbf{card}\{E_F\}$, the ordering function $h_D(\mathbf{v}) = -D(\mathbf{v}, \mathbf{I})$ will be maximal at the foreground and minimal at the background. Therefore, the morphological operator can be interpreted taking into consideration that in this case low values in $\mathcal{L}$ correspond to pixels close to the vector median or "background" and high values coincide with vector outliers or "foreground". It is important to remark that given an image parametrized for its spatial support, $\mathbf{I}_E : E \rightarrow \mathbb{R}^d$ and a subset in its spatial support, $Z \subset E$, for instance $\mathbf{I}_Z$ is a sector of $\mathbf{I}_E$, the depth functions associated of both images are not equivalent, i.e, $D(\cdot; \mathbf{I}_E) \neq D(\cdot; \mathbf{I}_Z)$. However, the *local knowledge* property [16] is preserved if and only if the depth function is calculated using the whole available image in E. The *erosion* of an



(a) $\mathbf{I}$ (b) $(\delta-\varepsilon)_{D_M,S}(\mathbf{I})$ (c) $(\delta-\varepsilon)_{D_P,S}(\mathbf{I})$ (d) $(\delta-\varepsilon)_{D_T,S}(\mathbf{I})$

(e) $\delta_{D_P,4S}(\mathbf{I})$ (f) $\varepsilon_{D_P,4S}(\mathbf{I})$ (g) $\delta_{D_P,8S}(\mathbf{I})$ (h) $\varepsilon_{D_P,8S}(\mathbf{I})$

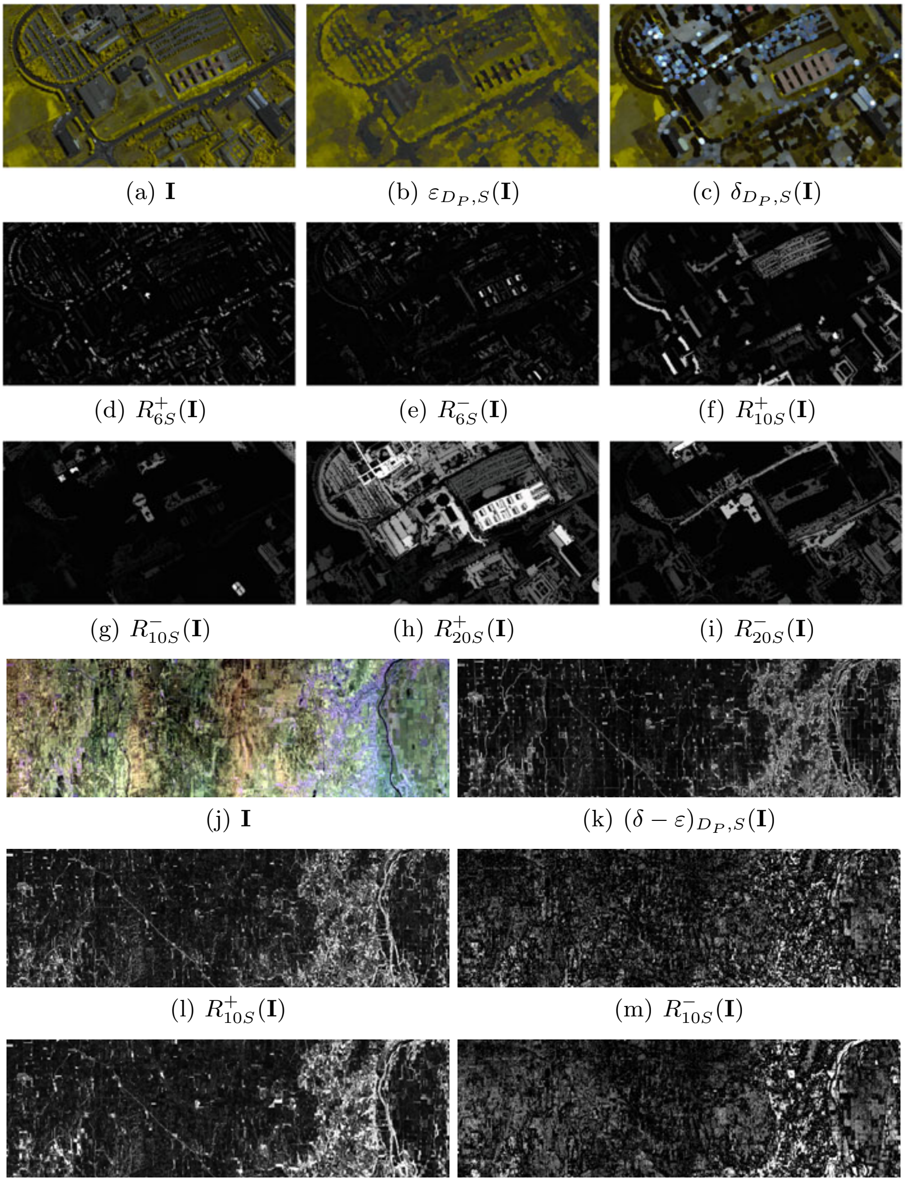
(i) $\mathbf{I}$ (j) $(\delta-\varepsilon)_{D_M,S}(\mathbf{I})$ (k) $(\delta-\varepsilon)_{D_P,S}(\mathbf{I})$ (l) $(\delta-\varepsilon)_{D_T,S}(\mathbf{I})$

(m) $\delta_{D_P,4S}(\mathbf{I})$ (n) $\varepsilon_{D_P,4S}(\mathbf{I})$ (o) $\delta_{D_P,8S}(\mathbf{I})$ (p) $\varepsilon_{D_P,8S}(\mathbf{I})$

(q) $\mathbf{I}$ (r) $(\delta-\varepsilon)_{D_M,S}(\mathbf{I})$ (s) $(\delta-\varepsilon)_{D_P,S}(\mathbf{I})$ (t) $(\delta-\varepsilon)_{D_T,S}(\mathbf{I})$

(u) $\delta_{D_M,4S}(\mathbf{I})$ (v) $\varepsilon_{D_M,4S}(\mathbf{I})$ (w) $\delta_{D_M,8S}(\mathbf{I})$ (x) $\varepsilon_{D_M,8S}(\mathbf{I})$

Fig. 4. Comparison of basic morphological operators using different depth functions. Note that false colors are not introduced by proposed approach. Gradients are calculated in depth function magnitude. (This figure should be viewed in color and by zooming on a computer screen.)



(a) $\mathbf{I}$

(b) $\varepsilon_{D_P,S}(\mathbf{I})$

(c) $\delta_{D_P,S}(\mathbf{I})$

(d) $R^{+}_{6S}(\mathbf{I})$

(e) $R^{-}_{6S}(\mathbf{I})$

(f) $R^{+}_{10S}(\mathbf{I})$

(g) $R^{-}_{10S}(\mathbf{I})$

(h) $R^{+}_{20S}(\mathbf{I})$

(i) $R^{-}_{20S}(\mathbf{I})$

(j) $\mathbf{I}$

(k) $(\delta - \varepsilon)_{D_P,S}(\mathbf{I})$

(l) $R^{+}_{10S}(\mathbf{I})$

(m) $R^{-}_{10S}(\mathbf{I})$

(n) $R^{+}_{20S}(\mathbf{I})$

(o) $R^{-}_{20S}(\mathbf{I})$

Fig. 5. Pavia University Hyperspectral Image (a-i). The original image has 610 × 340 pixels in 103 bands. Indian Pines Hyperspectral Image (j-o). The original image has 614 × 2678 pixels in 220 bands. Structure decomposition using residues of openings ($R^{+}_{\lambda S}$) and closings by reconstruction ($R^{-}_{\lambda S}$), calculate in depth function magnitude.

image $\mathbf{I}$ at pixel $\mathbf{v} \in \mathbb{R}^d$ by the structuring element $S \subset E$ using the h-depth function is given by $\varepsilon_{D,S}(\mathbf{I})(x) = \{\mathbf{I}(y) : \mathbf{I}(y) = \wedge_{h_D}[\mathbf{I}(z)], z \in S_x\}$, where $\wedge_{h_D}$ is the infimum according to the total ordering h_D induced for the depth function D. The corresponding adjunct *dilation* δ_D, is obtained by replacing the $\inf_{h_D}$ by the $\sup_{h_D}$, i.e., $\delta_{D,S}(\mathbf{I})(x) = \{\mathbf{I}(y) : \mathbf{I}(y) = \vee_{h_D}[\mathbf{I}(z)], z \in \hat{S}_x\}$, where $\hat{S}$ is the reflected structuring element. In practice, the erosion shrinks the structures which have a spectrum close to the outlier pixels in the vector dimensional space; "spatial peaks of spectra" thinner than the structuring element disappear by taking the spectrum of neighboring structures with a spectrum values close to the "background". As well, it expands the structures which have a vector value close to "foreground". Dilation produces the dual effects, enlarging the regions having a spectrum close to the outliers and contracting the background. The other morphological operators are defined and interpreted as product of this dilation and erosion. To illustrate the performance of our approach, we used the Berkeley Segmentation Dataset (BSDS) for examples of color images processing.Two examples for hyperspectral images are shown: (1) *University of Pavia* is an urban area that was recorded by the ROSIS-03 optical sensor and it has spatial dimensions of 610 by 340 pixels, with a spatial resolution of 1.3 meters per pixel. (2) *Indian Pine* obtained by the AVIRIS sensor in 220 spectral bands in the 400-250 nm range with 614×2678 pixels. In the experiments illustrated in Figs. 3, 4, and 5, the structuring element λS is a disk of size λ, and the number of random projections is $K = 2000$ for D_P. An optimal parameter selection can be done, however that is beyond the scope of this paper.

4 Conclusions

The paper proposed the statistical depth function as a powerful approach to induce a vector ordering for multivariate images and consequently a framework for unsupervised multivariate mathematical morphology. Indeed, it reaches a good compromise between simplicity and effectiveness in cases where no prior information is available for a supervised approach.

References

1. Al-Otum, H.M.: Morphological operators for color image processing based on mahalanobis distance. Optical Engineering 42(9), 2595–2606 (2003)
2. Angulo, J.: Morphological colour operators in totally ordered lattices based on distances: Application to image filtering, enhancement and analysis. Comput. Vis. Image Underst. 107(1-2), 56–73 (2007)
3. Aptoula, E., Lefèvre, S.: A comparative study on multivariate mathematical morphology. Pattern Recognition 40(11), 2914–2929 (2007)
4. Barnett, V.: The ordering of multivariate data (with discussion). Journal of the Royal Statistical Society Series A 139(3), 318–354 (1976)
5. Cuesta-Albertos, J.A., Nieto-Reyes, A.: The random tukey depth. Computational Statistics and Data Analysis 52, 4979–4988 (2008)

6. Donoho, D.L., Gasko, M.: Breakdown properties of location estimates based on halfspace depth and projected outlyingness. The Annals of Statistics 20(4), 1803–1827 (1992)
7. Garcia, A., Vachier, C., Vallée, J.P.: Multivariate mathematical morphology and bayesian classifier application to colour and medical images. In: Image Processing: Algorithms and Systems, p. 681203. SPIE, San Jose (2008)
8. Goutsias, J., Heijmans, H.J.A.M., Sivakumar, K.: Morphological operators for image sequences. Comput. Vis. Image Underst. 62(3), 326–346 (1995)
9. Lezoray, O., Elmoataz, A., Meurie, C.: Mathematical morphology in any color space. In: ICIAPW 2007: Proceedings of the 14th International Conference of Image Analysis and Processing - Workshops, pp. 183–187 (2007)
10. Liu, R.Y.: On a notion of data depth based on random simplices. Annals of Statistics 18, 405–414 (1990)
11. Mosler, K., Hoberg, R.: Data analysis and classification with the zonoid depth. DIMACS Series, pp. 49–59 (2006)
12. Najman, L., Talbot, H.: Mathematical morphology: from theory to applications. ISTE-Wiley (June 2010)
13. Pitas, I., Tsakalides, P.: Multivariate ordering in color image processing. IEEE Transactions on Circuits Systems Video Technology 1(3), 247–256 (1991)
14. Plaza, A., Martinez, P., Perez, R., Plaza, J.: A new approach to mixed pixel classification of hyperspectral imagery based on extended morphological profiles. Pattern Recognition 37(6), 1097–1116 (2004)
15. Serfling, R.: A depth function and a scale curve based on spatial quantiles. Statistical Data Analysis Based on the L_1 norm and Related Methods, 25–38 (2002)
16. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, Inc., Orlando (1983)
17. Serra, J.: The "False colour" problem. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 13–23. Springer, Heidelberg (2009)
18. Soille, P.: Morphological Image Analysis. Springer, Heidelberg (1999), http://web.ukonline.co.uk/soille
19. Tukey, J.W.: Mathematics and picturing data. In: Proceeding of the International Congress on Mathematics, pp. 523–531 (1975)
20. Velasco-Forero, S., Angulo, J.: Hit-or-miss transform in multivariate images. In: Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2010, Part I. LNCS, vol. 6474, pp. 452–462. Springer, Heidelberg (2010)
21. Velasco, S., Angulo, J.: Morphological processing of hyperspectral images using kriging-based supervised ordering. In: 17th IEEE International Conference on Image Processing (ICIP), pp. 1409–1412 (2010)
22. Vardi, Y., Zhang, C.H.: The multivariate l_1 median and associated data depth. Proceeding of the Nacional Academy of Sciences 97(4), 1423–1436 (2000)
23. Zuo, Y., Serfling, R.: General notions of statistical depth function. Annals of Statistics 28(2), 461–482 (2000)

Mathematical Morphology in Computer Graphics, Scientific Visualization and Visual Exploration

Jos B.T.M. Roerdink

Johann Bernoulli Institute for Mathematics and Computer Science
University of Groningen, P.O. Box 407, 9700 AK Groningen, The Netherlands
j.b.t.m.roerdink@rug.nl
http://www.cs.rug.nl/svcg

Abstract. Historically, mathematical morphology has primarily focused on the processing and analysis of two-dimensional image data. In this paper, we survey a number of other areas where mathematical morphology finds fruitful application, such as computer graphics and solid modeling; path planning; filtering, segmentation and visualization of volume data; or visual exploration of high-dimensional data. We also mention techniques for accelerating morphological computations by using graphics hardware (GPU computing).

Keywords: Constructive solid geometry, computational geometry, group morphology, volume processing and visualization, high-dimensional data exploration, GPU computing.

1 Introduction

Historically, mathematical morphology originated as a methodology for processing and analyzing two-dimensional image data. However, its scope turned out to be much wider, being applicable to general multi-dimensional data. In this paper we review a number of such areas which go beyond the image analysis domain. In constructive solid geometry, Minkowski operators are used for modelling and visualization of 3D objects. Group morphology is applicable to path planning and configuration space analysis. Morphological operators have been used for transfer function design in volume rendering. Morphological pyramids and connected morphological operators find application in multiresolution visualization and filtering of (medical) volume data, while for volumetric segmentation morphological active surface models have been proposed.

More recently, mathematical morphology has been applied in visual exploration of high-dimensional data. For example, the watershed transform has been adapted for fast reconstruction and visualization of brain networks; connected filters are used for finding relevant subspaces in high-dimensional scientific data, or for filtering tensor fields such as diffusion tensor imaging data. We also briefly discuss recent techniques for accelerating morphological computations by using graphics hardware (GPU computing).

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 367–380, 2011.

2 Computer Graphics and Computational Geometry

2.1 Computer Graphics

In computer graphics various techniques are used to synthesize realistic images of real-world scenes. In Constructive Solid Geometry (CSG), composite solid objects are formed by applying set operations (union, intersection and difference) to simple solids, such as spheres, blocks, or cylinders. Visualization of these objects is possible by incorporating the CSG operations in the ray tracing technique, which visualizes 3D objects by simulating the physical processes of ray propagation, reflection and transmission [11, 13].

Various authors have found the Minkowski addition and subtraction operators to be suitable tools in CSG [30, 40]. Other uses of Minkowski operations in shape description and solid modeling were presented by Ghosh [12]. In special cases the Minkowski addition reduces to the sweep representation (a special case of translation surface) in solid modeling [11, 13].

A different approach was followed in [35]. Using a basic set of elementary shapes, a decomposition was derived of a multiple Minkowski sum of any number of objects, chosen from the basic set, into a union of standard primitives. This union can then subsequently be visualized by standard CSG combined with ray tracing; see Figure 1 for an example. The advantages of this method are efficiency (once the decomposition has been carried out, the ray tracing process is comparatively fast) and compactness of the representation.

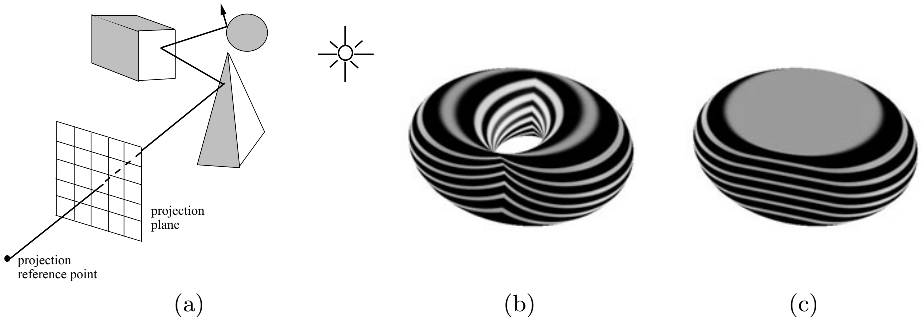


Fig. 1. (a): Set-up for ray tracing of a 3D scene. Visualization of the Minkowski sum of: (b) two perpendicular circles, a quartic surface defined by the equation $(x^2 - y^2 + z^2 + r_z^2 - r_y^2)^2 = 4x^2(r_z^2 - y^2)$; (c) two perpendicular flat discs [35].

2.2 Path Planning and Configuration Space Analysis

In path planning the problem is to find a path for an object, say a robot or a car, moving in a space with obstacles. The problem falls apart into two distinct subproblems [15]. First, the *empty-space problem*: find the allowed states of the robot (moving object); second, the *find-path problem*: find a trajectory in the empty space, subject to certain constraints.

Morphological operations can be used to address the empty-space problem. For robots with only translational degrees of freedom, one can find the allowed positions of the (arbitrarily chosen) center of the robot by a standard erosion of the space outside the obstacles, where the structuring element B is the robot itself. Equivalently, one may perform the dilation by the reflected set $\check{B}$ of the set of obstacles to find the *forbidden* positions of the center of the robot. This is more efficient when the obstacle space is smaller than the space outside the obstacles. If the robot has rotational degrees of freedom or rotating joints, the framework of *group morphology* is appropriate.

Group morphology. The theory of group morphology deals with the construction of morphological operators on an homogeneous space $(\mathbb{T}, \mathcal{X})$, where $\mathbb{T}$ is a group acting transitively on $\mathcal{X}$. For background, see Heijmans and Ronse [17, 38] for the case of *abelian* symmetry groups, and Roerdink [31, 34] for the case of arbitrary (abelian and non-abelian) symmetry groups; see also [19]. For example, when $\mathcal{X} = \mathbb{R}^d$ any appropriate group $\mathbb{T}$ may be chosen, such as the translation group, the motion group, the affine group, or the projective group, where in each case the morphological operations of interest are invariant under the corresponding group $\mathbb{T}$.

As shown in [32], translation-rotation morphology, where $\mathbb{T}$ is the motion group, is applicable to the empty space problem for robots with rotational degrees of freedom. Another application is the *tailor problem*, which concerns the fitting of sets without overlap within a larger set [33].

An interesting approach to configuration space analysis and similar problems was presented recently by Lysenko *et al.*, who reformulated the framework of group morphology in terms of *group convolution algebras* [27].

2.3 Shape Comparison and Symmetry Detection

Shape comparison is one of the fundamental problems of machine vision. For the case of convex polygons and convex polyhedra shape similarity measures have been studied based on Minkowski addition and inequalities related to the Brunn-Minkowski theory [18, 47]. The same theory can be applied for symmetry detection of convex polyhedra, see for example [48]. Also, group convolution algebras have been applied for this purpose [27].

If one considers measures which are not only translation-invariant, but also invariant under the group of orthogonal transformations, the direct computation of similarity measures in the 3D case becomes very time consuming. In principle, optimization should be performed for all possible positions of rotation axes and rotation angles. As shown in [47], for certain measures based on (mixed) volume, it is sufficient to consider only a finite number of “critical” rotations; see also [1]. By using geometric inequalities in the slope diagrams of the polyhedra, the set of relative orientations to be considered can be narrowed down, so that the time complexity of $O(n^6)$ is reduced to $O(n^{4.5})$ [37].

3 Volume Processing and Visualization

Volume visualization, or volume rendering, is a technique which produces two-dimensional image representations of three-dimensional data from different viewpoints, using computer graphics techniques such as illumination, shading and colour [16]. Two types of rendering are distinguished: (i) *surface rendering*, where the volume is reduced to one or more isosurfaces $S(c) : f(x, y, z) = c$ of a density function f representing the boundary between materials; (ii) *direct volume rendering*, which maps the volume data directly on the screen, with semi-transparent effects (see Figure 2). Two volume rendering methods which are widely used in medical imaging are *X-ray rendering* and *maximum intensity projection* (MIP). Here one generates, for each pixel of the view plane, a ray through the data parallel to the line of sight (i.e., perpendicular to the view plane), and assigns either the average or the maximum data value encountered along this ray to the pixel. Because of its computational simplicity and effectiveness, MIP is widely used in the display of magnetic resonance angiography (MRA) and ultrasound data.

Especially when interactive rendering rates are required (i.e., there must be a fast response of the rendering system to actions of the user), volume rendering is a very demanding problem when the sizes of the volume data are large. Two general approaches are available for accelerating the involved computations.

- *Special hardware.* In this category we will briefly discuss general purpose computation on graphics hardware.
- *Special data structures.* Hierarchical structures like wavelets or morphological pyramids are of special interest here.

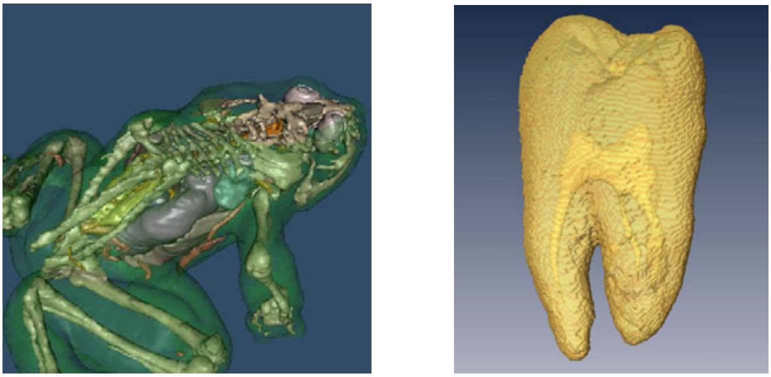


Fig. 2. Left: surface rendering of a frog data set with partially transparent surfaces corresponding to different tissues (data source: the VTK distribution [42]). Right: direct volume rendering of tooth data (data source: the Volume Library http://www9.informatik.uni-erlangen.de/External/vollib)

3.1 General Purpose Computation on Graphics Hardware

A recent development which has a major impact on interactive data processing and visualization, is new programmable graphics hardware. The Graphics Processing Unit (GPU), originally used for graphical tasks only, has evolved into a so-called General Purpose GPU (GPGPU). The computing power of GPUs is currently increasing at a faster pace than that of CPUs, so that the GPU is now a major computational device for diverse applications, such as physics simulations, neural networks, image processing and computer vision, graphics and visualization, and even database sorting.

Initially, GPGPU applications, even though not concerned with graphics rendering, did have to use the rendering paradigm, involving the use of textures. As an example we mention the GPU-acceleration of elementary morphological operators as pioneered in the nineties by Hopf and Ertl [20]. However, in 2006 NVidia introduced a programming environment called CUDA [25], which allows the GPU to be programmed through more traditional means. At this moment a dedicated programming effort is still required to develop algorithms that perform efficiently on GPU hardware, but efforts are underway for automatic transformation of CPU programs into GPU counterparts [24].

3.2 Morphological Operators for Transfer Function Design

In volume rendering, one of the most difficult tasks is the process of *classification*, that is, to determine for each voxel to what type of material, tissue, etc., it belongs. Classification is usually done by using a *transfer function* that defines the colour and opacity values of each voxel. These are then used when the values of voxels along viewing rays are combined into a single pixel color. Finding a suitable transfer function is often done by an interactive process which can be very complicated and time consuming [16, Ch. 9]. To make this process more automatic, Lürig and Ertl [26] proposed multiscale morphological operators which incorporate spatial neighbourhood information, as an alternative to traditional transfer functions.

3.3 Morphological Pyramids for Multiresolution Visualization

For very large data sets, a multiresolution approach is an obvious choice, which allows a quick visualization of reduced versions of the data that can be progressively refined if needed. For maximum intensity projection (MIP), the transform is nonlinear, so the standard linear multiresolution models based on wavelets (see, e.g., [50]) are not applicable. Instead, the framework of morphological pyramids as developed by Goutsias and Heijmans [14] can be used as the basis for developing multiresolution algorithms for MIP; see [36] for a survey.

The multiresolution MIP algorithm can be summarized as follows. In the *pyramid analysis* phase, which is a preprocessing step, a 3D morphological pyramid of approximation and detail coefficients is computed by repeated morphological filtering followed by downsampling. The original data at each level of the pyramid can be recovered by successive upsampling and morphological filtering of higher-level data; this process is called *pyramid synthesis*. Particularly suitable for multiresolution MIP are *dilation pyramids*, where the filter operation during synthesis is a dilation. For such pyramids, the MIP operation and the pyramid synthesis work nicely together, in the sense that the maxima along the line of sight can be computed first from pyramid data on a coarse level (where the size of the data is reduced), after which a fast 2D morphological synthesis operator is used to perform reconstruction of the projection image to full grid resolution [36].

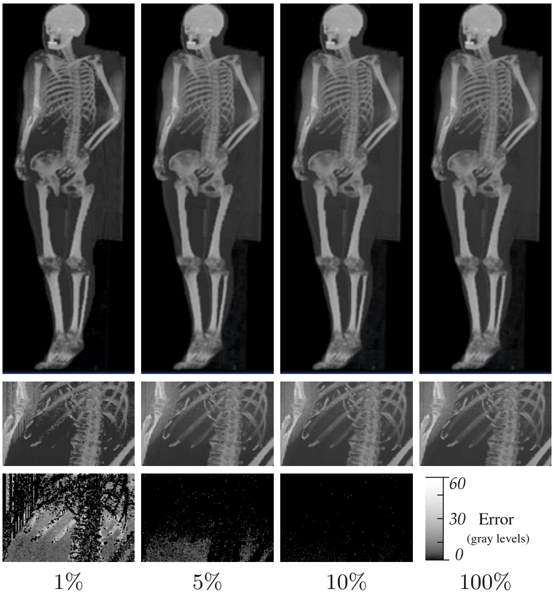


Fig. 3. Streaming MIP-splatting of the complete Visible Woman dataset (source: http://www.nlm.nih.gov/research/visible) in a 800 by 2000 window. The rendering is shown at various quality settings (given as percentages of the total number of detail coefficients). The second row shows a detail image for each quality setting, and the third row shows the difference image in gray levels [23].

Several approaches based on pyramid schemes (all of the dilation-pyramid type) were compared in [36]. The most efficient approach was found to be *streaming MIP-splatting*. In this method, detail coefficients from all levels are jointly resorted with respect to decreasing magnitude of a suitable error measure. In the rendering phase, all resorted coefficients are projected successively, until a desired accuracy is obtained. As shown in [36], streaming MIP-splatting outperforms earlier methods based on morphological pyramids, both with respect to image quality with a fixed amount of detail data, and in terms of a flexible trade-off between approximation error and computing time.

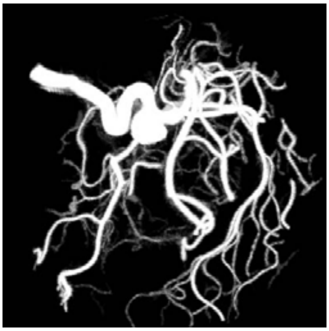


Fig. 4. MIP rendering of the aneurysm data set (rotational b-plane X-ray scan, courtesy Philips Research, Hamburg, Germany; `http://volvis.org`), filtered according to a non-compactness criterion with value $\lambda = 2.0$ [51]

Streaming MIP-splatting on the GPU. A GPU implementation of the streaming MIP algorithm was studied in [23]. The load and the dataset can be spread over multiple graphics cards in a straightforward way, thereby achieving support for large volume data with an almost optimal speedup. An example is given in Figure 3. The method achieves interactive frame rates, ranging from 20-50 frames per second, depending on the allowed error.

3.4 Connected Operators for Combined Filtering and Visualization

Connected filters are based upon an axiomatic definition of connectivity within a complete lattice framework [19, 43]. They are used to perform filtering based on various shape and size attributes. A key property of connected filters is their edge preserving nature. Connected filters can be efficiently computed by the Max-tree data structure, in which the nodes represent connected components for all threshold levels in a data set [41]. The basic Max-tree data structure can be augmented by extensions that allow (i) direct volume rendering, (ii) representation of the Max-tree on graphics hardware, and (iii) fast active cell selection for isosurface generation. In all three cases, the Max-tree representation can be used to change filter parameters interactively and visualize the result at interactive rates [51]; see Figure 4 for an example.

3.5 Segmentation and Visualization by Active Surface Models

Segmenting images using active contour models (snakes) involves the evolution of a curve or surface (*i.e.*, an interface), subject to constraints derived from a given input image or volume. State-of-the-art active contours are based on the level set framework, which is able to handle complicated topologies of the underlying shapes [46]. The evolving curve $C(s) : [0,1] \to \mathbb{R}^2$ is given by the zero level-set at time t of a function $\phi(x, y, t)$ that satisfies an evolution equation of the form $\frac{\partial \phi}{\partial t} = F \, ||\nabla \phi||$. In the context of image segmentation, various formulations

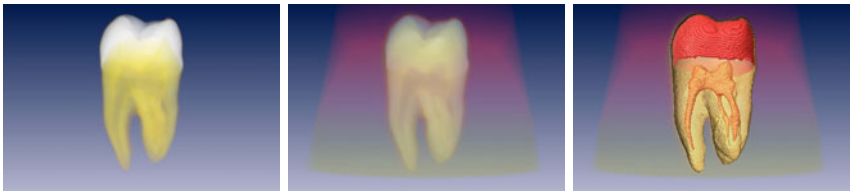


Fig. 5. Segmentation of multiple nested objects. *Left, center*: volume renderings of the tooth data set (cf. Figure 2); *right*: the segmented constituent parts: the enamel, dentin and root canal [21].

for the speed function F have been proposed. Traditionally F is set to some function of the gradient image [6, 28], such that the active contour stops its evolution whenever important edges in the input image are encountered. Chan and Vese [7] used a minimum-variance criterion of the segmented regions. Their active contour model leads to the evolution equation:

$$\frac{\partial \phi}{\partial t} = \left\{ \mu \cdot \kappa - \nu + \lambda \left[(I - c_2)^2 - (I - c_1)^2 \right] \right\} ||\nabla \phi|| \,, \tag{1}$$

which has to be solved for ϕ, with I the input image, κ the level set curvature, $c_1 = \text{average}(I)$ in $\{\phi \geq 0\}$ and $c_2 = \text{average}(I)$ in $\{\phi < 0\}$. The first term in Eq. (1) represents the curvature flow and minimizes the length of the curve, the second term represents inwards motion at constant speed and minimizes the area of the region, whereas the last term represents region competition by the minimum-variance criterion.

As shown in [21], Chan and Vese's minimum-variance model can be reformulated within the context of discrete *multi-scale morphology* [29] as follows:

$$u^{k+1} = \text{sgn}\left(u^k * \chi_{B_p} + \text{sgn}(f^k) \left(|B_p| - 1 \right) \right) , \tag{2}$$

where k is the discrete time (scale) parameter, $\text{sgn}(x) = 1$ if $x > 0$ and -1 otherwise, B_p is the unit ball w.r.t. the p-norm, χ_{B_p} the characteristic function of B_p, $|B_p|$ the number of elements of B_p, and '$*$' denotes linear convolution. The 'speed function' f^k is given by

$$f^k = \lambda \left((I - c_2^k)^2 - (I - c_1^k)^2 \right) + \alpha \cdot \text{sgn}(u^k * \chi_{B_p} + \beta), \tag{3}$$

where $\lambda \geq 0$, $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{Z}$.

By varying the free parameters in the definition of f^k, various filters are obtained, such as median filtering, local dilations and erosions, or open-close and close-open filters. Both the PDE-based model and the discrete model were implemented on the GPU in [21]. Experiments showed that the discrete model produces results which are comparable to those of the continuous PDE model based on level sets, while being almost two orders of magnitude faster. An example is given in Figure 5, showing the segmentation of multiple nested objects.

4 Visual Exploration of High-Dimensional Data

4.1 Reconstruction and Visualization of Brain Networks

Electroencephalography (EEG) is a method to measure the electrical activity of the brain by means of electrodes attached to the scalp at multiple locations. Synchronous electrical activity in different brain regions is generally assumed to imply functional relationships between these regions. A measure for this synchrony is EEG coherence, calculated between pairs of electrode signals as a function of frequency.

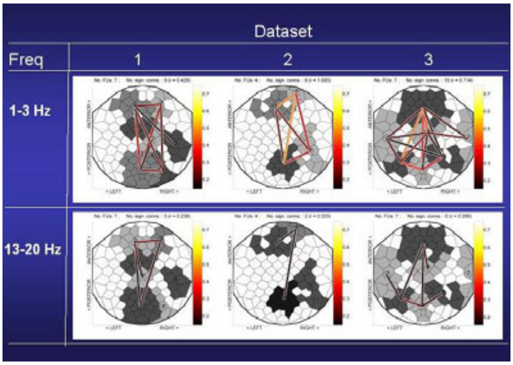


Fig. 6. *Functional Unit maps for multichannel EEG coherence visualization.* Brain responses were collected from three subjects using an EEG cap with 119 scalp electrodes. During a so-called P300 experiment, each participant was instructed to count target tones of 2000 Hz (probability 0.15), alternated with standard tones of 1000 Hz (probability 0.85) which were to be ignored. After the experiment, the participant had to report the number of perceived target tones. To each electrode a Voronoi cell in a graph layout is associated and all cells belonging to a functional unit (FU) have a corresponding color. Lines connect FU centers if the inter-FU coherence exceeds a significance threshold. The color of a line depends on the inter-FU coherence. Shown are FU maps for target stimuli data, with FUs larger than 5 cells, for the 1-3Hz EEG frequency band (top row) and for 13-20Hz (bottom row), for three datasets [5].

A typical data-driven visualization of electroencephalography (EEG) coherence is a graph layout, with vertices representing electrodes and edges representing significant coherences between electrode signals. A drawback of this layout is its visual clutter when the number of electrodes is large. To reduce clutter, ten Caat *et al.* [5] defined a so-called *functional unit* (FU) as a data-driven region of interest (ROI). An FU is a spatially connected set of electrodes recording pairwise significantly coherent signals, represented in the coherence graph by a spatially connected clique. Computing such cliques is very time-consuming: its time complexity is $O(3^{n/3})$, with n the number of vertices.

As an alternative, a modified watershed method (time complexity $O(n^2 \log n)$) was developed, which merges basins representing FUs during the segmentation process if they are spatially connected and if their union is a clique [5]. The modified watershed method produces FU maps which are comparable to the clique-based method, and is up to a factor of 10^5 faster for a typical setting with 128 EEG channels, thus making interactive visualization possible; see Figure 6 for an example. The method can also be extended to find averaged maps for data-driven group analysis. The method was applied to mental fatigue [4] and neurodegenerative disease [8].

4.2 Filtering and Visualization of Diffusion Tensor Imaging Data

Processing and visualization of tensor fields has become very important over the last decade [49]. A prime application area is medical imaging, where magnetic resonance diffusion tensor imaging (DTI) enables the *in vivo* exploration of the structural organization of fibrous tissue, such as the brain or the heart. An interesting feature of DTI is its ability to derive local information, such as the amount of anisotropy in a single brain voxel; this can be visualized by tensor glyphs [22], which represent iso-probability surfaces of the diffusion process (in ordinary DTI these are ellipsoids oriented along the main fiber direction). In addition, one can track fiber bundles from a selected brain area. This allows the determination and visualization of *structural connectivity* between brain regions; see Fig. 7 for an example [9].

For connectivity-based morphological filtering and visualization of tensor fields, new developments in (hyper)connectivity, constrained and partial connectivity are of current interest [3, 39, 44, 45, 52].

4.3 Finding Relevant Subspaces in High-Dimensional Data

Data sets in many scientific areas are growing to enormous sizes with high dimensionality. Exploration of such large data spaces poses a huge challenge. *Subspace clustering* is one among several approaches which have been proposed for this

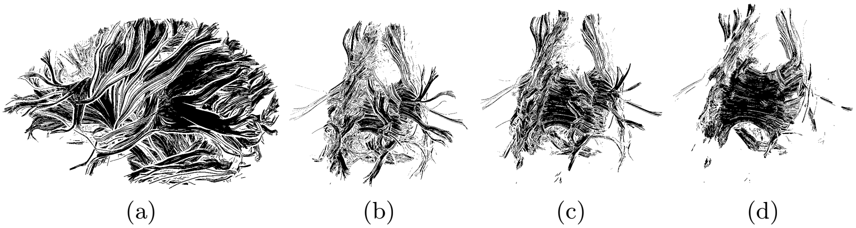


Fig. 7. Illustrative visualization of DTI fiber tracts. (a): Initial set. (b)-(d): Three example stages of filtering, using the fractional anisotropy (FA) value of a DTI fiber tract data set. With the growing filtering threshold for FA, more of the internal structure of the data set is revealed [9].

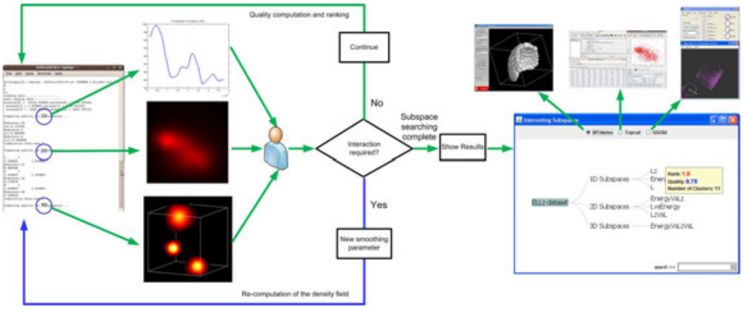


Fig. 8. Schematic diagram of the interactive search and exploration system of high-dimensional data spaces [10]

purpose in recent years. This method detects subspaces of a high-dimensional space which are potentially "relevant" or "interesting", as defined by specific criteria. The actual clustering can then be limited to the relevant subspaces. Ferdosi *et al.* [10] recently proposed a subspace finding method based on connected morphological operators. First a transformation is performed from the high-dimensional parametric space to discrete image space where the data are represented by a grid-based density field. Then connected operators are applied on this density field that provides visual support for the analysis of the important subspaces. The importance of a cluster is measured by a quality criterion based upon the notion of *dynamics* [2]. The search for modes/local maxima is done on the Max-tree representation of the density image [41]. For subspaces of dimension higher than three, principal component analysis (PCA) is applied and the first three principal components are used for subspace ranking.

During computation the user can interact with the system to improve the results. In the result stage, three visualization toolkits are used that are linked within a graphical user interface for in-depth exploration of the ranked subspaces; see Figure 8, where the system is used in an astronomical application. Current work involves the extension of this approach to large touch-sensitive displays, which support collaborative research.

5 Conclusions

As is apparent from this brief survey, mathematical morphology is a very versatile methodology, with applications ranging from image processing and computer graphics to data visualization. Built on a solid mathematical foundation, it continues to find new theoretical directions such as (hyper)connected filters, as well as important applications, including tensor imaging and high-dimensional data exploration.

References

1. Bekker, H., Brink, A.A., Roerdink, J.B.T.M.: Reducing the time complexity and identifying ill-posed problem instances of Minkowski sum based similarity calculations. International Journal of Computational Geometry and Applications 19(5), 441–456 (2009)
2. Bertrand, G.: On the dynamics. Image Vision Comput. 25(4), 447–454 (2007)
3. Braga-Neto, U., Goutsias, J.: A theoretical tour of connectivity in image processing and analysis. J. Math. Imag. Vision 19, 5–31 (2003)
4. ten Caat, M., Lorist, M.M., Bezdan, E., Roerdink, J.B.T.M., Maurits, N.M.: High-density EEG coherence analysis using functional units applied to mental fatigue. J. Neuroscience Methods 171(2), 271–278 (2008), doi:10.1016/j.jneumeth.2008.03.022
5. ten Caat, M., Maurits, N.M., Roerdink, J.B.T.M.: Data-driven visualization and group analysis of multichannel EEG coherence with functional units. IEEE Trans. Visualization and Computer Graphics 14(4), 756–771 (2008)
6. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. 5th Int. Conf. Computer Vision, pp. 694–699 (1995)
7. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. Image Processing 10, 266–277 (2001)
8. Crippa, A., Maurits, N.M., Roerdink, J.B.T.M.: Graph averaging as a means to compare multichannel EEG coherence networks and its application to the study of mental fatigue and neurodegenerative disease. Computers & Graphics 35(2), 265–274 (2011)
9. Everts, M.H., Bekker, H., Roerdink, J.B.T.M., Isenberg, T.: Depth-Dependent Halos: Illustrative Rendering of Dense Line Data. IEEE Transactions on Visualization and Computer Graphics 15(6), 1299–1306 (2009)
10. Ferdosi, B.J., Buddelmeijer, H., Trager, S., Wilkinson, M.H.F., Roerdink, J.B.T.M.: Finding and visualizing relevant subspaces for clustering high-dimensional astronomical data using connected morphological operators. In: Proceedings of IEEE Conference on Visual Analytics Science and Technology (IEEE VAST), pp. 35–42 (October 2010)
11. Foley, J.D., Dam, A.V., Feiner, S.K.: Computer Graphics: Principles and Practice. Addison-Wesley, Reading (1990)
12. Ghosh, P.K.: A mathematical model for shape description using Minkowski operators. Comp. Vis. Graph. Im. Proc. 44, 239–269 (1988)
13. Glassner, A.S. (ed.): An Introduction to Ray Tracing. Academic Press, New York (1989)
14. Goutsias, J., Heijmans, H.J.A.M.: Multiresolution signal decomposition schemes. Part 1: Linear and morphological pyramids. IEEE Trans. Image Processing 9(11), 1862–1876 (2000)
15. Gouzènes, L.: Strategies for solving collision-free trajectories problems for mobile and manipulator robots. Intern. J. Robotics Res. 3, 51–65 (1984)
16. Hansen, C.D., Johnson, C.R. (eds.): The Visualization Handbook. Elsevier, Butterworth-Heinemann (2005)
17. Heijmans, H.J.A.M., Ronse, C.: he algebraic basis of mathematical morphology. Part I: dilations and erosions. Comp. Vis. Graph. Im. Proc. 50, 245–295 (1989)
18. Heijmans, H.J.A.M., Tuzikov, A.: Similarity and symmetry measures for convex shapes using Minkowski addition. IEEE Trans. Patt. Anal. Mach. Intell. 20(9), 980–993 (1998)

19. Heijmans, H.J.A.M.: Morphological Image Operators. Advances in Electronics and Electron Physics 25(suppl.) (1994)
20. Hopf, M., Ertl, T.: Accelerating Morphological Analysis with Graphics Hardware. In: Workshop on Vision, Modelling, and Visualization VMV 2000, pp. 337–345 (2000)
21. Jalba, A.C., Roerdink, J.B.T.M.: An efficient morphological active surface model for volumetric image segmentation. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) Proc. 9th International Symposium on Mathematical Morphology and its Application to Signal and Image Processing, Groningen, August 24-27, pp. 193–204 (2009)
22. Kindlmann, G.L., Westin, C.F.: Diffusion tensor visualization with glyph packing. IEEE Trans. Vis. Comput. Graph 12(5), 1329–1336 (2006)
23. van der Laan, W.J., Jalba, A.C., Roerdink, J.B.T.M.: Multiresolution MIP rendering of large volumetric data accelerated on graphics hardware. In: Proc. Eurographics/IEEE VGTC Symposium on Visualization (EuroVis), pp. 243–250 (2007)
24. Leung, A., Lhoták, O., Lashari, G.: Automatic parallelization for graphics processing units. In: Proceedings of the 7th International Conference on Principles and Practice of Programming in Java, PPPJ 2009, pp. 91–100. ACM, New York (2009)
25. Lindholm, E., Nickolls, J., Oberman, S., Montrym, J.: NVIDIA Tesla: A unified graphics and computing architecture. IEEE Micro. 28(2), 39–55 (2008)
26. Lürig, C., Ertl, T.: Hierarchical volume analysis and visualization based on morphological operators. In: Proc. IEEE Visualization 1998, pp. 335–341. IEEE Computer Society Press, Los Alamitos (1998)
27. Lysenko, M., Nelaturi, S., Shapiro, V.: Group morphology with convolution algebras. In: Proceedings of the 14th ACM Symposium on Solid and Physical Modeling, SPM 2010, pp. 11–22. ACM, New York (2010)
28. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: A level set approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 158–175 (1995)
29. Maragos, P.: Differential morphology and image processing. IEEE Transactions on Image Processing 5(6), 922–937 (1996)
30. Menon, J.P., Marisa, R.J., Zagajac, J.: More powerful solid modeling through ray representations. IEEE Computer Graphics and Applications 14(3), 22–35 (1994)
31. Roerdink, J.B.T.M.: Mathematical morphology with non-commutative symmetry groups. In: Dougherty, E.R. (ed.) Mathematical Morphology in Image Processing, ch. 7, pp. 205–254. Marcel Dekker, New York (1993)
32. Roerdink, J.B.T.M.: Solving the empty space problem in robot path planning by mathematical morphology. In: Serra, J., Salembier, P. (eds.) Proc. Workshop 'Mathematical Morphology and its Applications to Signal Processing', Barcelona, Spain, May 12-14, pp. 216–221 (1993)
33. Roerdink, J.B.T.M.: The generalized tailor problem. In: Maragos, P., Shafer, R.W., Butt, M.A. (eds.) Mathematical Morphology and its Applications to Image and Signal Processing, pp. 57–64. Kluwer Acad. Publ., Dordrecht (1996)
34. Roerdink, J.B.T.M.: Group morphology. Pattern Recognition 33(6), 877–895 (2000)
35. Roerdink, J.B.T.M., Blaauwgeers, G.S.M.: Visualization of Minkowski operations by computer graphics techniques. In: Serra, J., Soille, P. (eds.) Mathematical Morphology and its Applications to Image Processing, pp. 289–296. Kluwer Acad. Publ., Dordrecht (1994)

36. Roerdink, J.B.T.M.: Morphological pyramids in multiresolution MIP rendering of large volume data: Survey and new results. J. Math. Imag. Vision 22(2/3), 143–157 (2005)
37. Roerdink, J.B.T.M., Bekker, H.: Similarity measure computation of convex polyhedra revisited. In: Bertrand, G., Imiya, A., Klette, R. (eds.) Digital and Image Geometry. LNCS, vol. 2243, pp. 389–399. Springer, Heidelberg (2002)
38. Ronse, C., Heijmans, H.J.A.M.: The algebraic basis of mathematical morphology. Part II: openings and closings. Comp. Vis. Graph. Im. Proc.: Image Understanding 54, 74–97 (1991)
39. Ronse, C.: Partial partitions, partial connections and connective segmentation. Journal of Mathematical Imaging and Vision 32(2), 97–125 (2008)
40. Rossignac, J.R., Requicha, A.A.G.: Offsetting operations in solid modeling. Computer Aided Geometric Design 3, 129–148 (1986)
41. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Transactions on Image Processing 7, 555–570 (1998)
42. Schroeder, W., Martin, K., Lorensen, B.: The Visualization Toolkit: an Object Oriented Approach to 3D Graphics, 4th edn. Kitware, Inc. (2006)
43. Serra, J. (ed.): Image Analysis and Mathematical Morphology. Theoretical Advances, vol. II. Academic Press, New York (1988)
44. Serra, J.: Connectivity on complete lattices. J. Math. Imag. Vision 9(3), 231–251 (1998)
45. Soille, P.: Constrained connectivity for hierarchical image decomposition and simplification. IEEE Trans. Pattern Anal. Mach. Intell. 30(7), 1132–1145 (2008)
46. Suri, J., Liu, K., Singh, S., Laxminarayan, S., Zeng, X., Reden, L.: Shape recovery algorithms using level sets in 2-D/3-D medical imagery: A state of the art review. IEEE Trans. on Inf. Tech. in Biomed. 6, 8–28 (2002)
47. Tuzikov, A.V., Roerdink, J.B.T.M., Heijmans, H.J.A.M.: Similarity measures for convex polyhedra based on Minkowski addition. Pattern Recognition 33(6), 979–995 (2000)
48. Tuzikov, A.V., Sheynin, S.: Symmetry measure computation for convex polyhedra. Journal of Mathematical Imaging and Vision 16(1), 41–56 (2002)
49. Weickert, J., Hagen, H. (eds.): Visualization and Processing of Tensor Fields. Springer, Berlin (2006)
50. Westenberg, M.A., Roerdink, J.B.T.M.: Frequency domain volume rendering by the wavelet X-ray transform. IEEE Trans. Image Processing 9(7), 1249–1261 (2000)
51. Westenberg, M.A., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Volumetric attribute filtering and interactive visualization using the Max-tree representation. IEEE Trans. Image Processing 16(12), 2943–2952 (2007)
52. Wilkinson, M.H.F.: Attribute-space connectivity and connected filters. Image and Vision Computing 25, 426–435 (2007)

Surface Reconstruction Using Power Watershed

Camille Couprie[1], Xavier Bresson[2], Laurent Najman[1],
Hugues Talbot[1], and Leo Grady[3]

[1] Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Equipe A3SI,
ESIEE Paris (93160 Noisy-le-Grand, France)
[2] City University of Hong Kong, Dpt. of Computer Science (Hong Kong)
[3] Siemens Corporate Research,
Dpt. Imaging Analytics & Informatics (Princeton USA)
{c.couprie,l.najman,h.talbot}@esiee.fr,
xbresson@cityu.edu.hk, leo.grady@siemens.com

Abstract. Surface reconstruction from a set of noisy point measurements has been a well studied problem for several decades. Recently, variational and discrete optimization approaches have been applied to solve it, demonstrating good robustness to outliers thanks to a global energy minimization scheme. In this work, we use a recent approach embedding several optimization algorithms into a common framework named power watershed. We derive a specific watershed algorithm for surface reconstruction which is fast, robust to markers placement, and produces smooth surfaces. Experiments also show that our proposed algorithm compares favorably in terms of speed, memory requirement and accuracy with existing algorithms.

Keywords: optimization, point measurements, Graph cuts, total variation.

1 Introduction

This paper develops a watershed-based algorithm providing a global optimal solution to the surface reconstruction problem from a set of scattered points. Surface fitting is a challenging problem when dealing with data containing sparse noise, gaps, and outliers. The set of points may be for example acquired by several scans of an object (range scanning). In this context, regularization-based methods have been shown to be robust when the points are lacking connectivity, ordering information, and may be contaminated by noise. While there exist numerous explicit surface extraction techniques that estimate the exact positions of surface points, in this work we will focus on implicit surface representation. Implicit surfaces may be represented by level sets (*e.g.* [29]) or binary partitions (*e.g.* [23]).

Local methods for surface reconstruction including the MPU method [5] are sensitive to noise, as shown in the experiments of [19]. Among the recent global approaches, the Poisson method [20] and FFT-method [21] are more robust to noise, however they require orientation information.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 381–392, 2011.

The method of Jalba and Roerdink [19] makes it possible to avoid having to estimate orientation information. This is achieved by computing approximations of Coulomb potentials in a grid as an input for the convection method of [29]. We propose a different approach that allows us to perform a global optimization of the surface reconstruction problem without requiring the computation of such field as Coulomb potentials.

A generic optimization-based regularization formulation for shape fitting minimizes the total variation functional weighted by the distance function from the set of points P. More generally, given two positive numbers p and q, we consider an object indicator partition u solution of

$$\begin{aligned} &\min_{u\in[0,1]} \int_{\Omega} w(z)^p |\nabla u(z)|^q \mathrm{dz} \\ &\text{subject to } u(z) = 0 \;\; \forall z \in \Omega_{\text{in}}, \\ &\text{and } u(z) = 1 \;\; \forall z \in \Omega_{\text{out}}, \end{aligned} \tag{1}$$

where Ω_{in} is the set of labels inside the surface and Ω_{out} is the set of labels outside the surface. The weight function w is defined at every point z in a grid as $w(z) = d_P(z)$, where $d_P(z)$ is the distance map from the points. When p is finite and $q = 1$, Eq. (1) leads to a binary solution u [27]. A solution can be deduced in the discrete setting using *e.g.* the network flow technique [12], also known as Graph cuts [4]. Augmenting path max flow implementations are fast and efficient in 2D but memory consuming as the connectivity increases, for instance in 3D. Lempitzky and Boykov [23] have overcome this problem by limiting the size of the search for a solution in the grid while still guaranteeing a global optimum. However their solution is based on restrictive assumptions assuming that the data points are provided with an estimate of the surface orientation. Furthermore, at low resolution, results exhibit metrication artifacts and look blocky, so a high resolution is essential for getting smooth results using the Graph cuts method.

We propose a watershed-based approach in order to provide a way to quickly obtain smooth surfaces at a high resolution without any need to pre-estimate the surface orientation. Recently, Couprie *et al.* introduced the power watershed method [10,8], which can be seen as an anisotropic discretization of (1) with $p \rightarrow \infty$ and $q = 2$. Although this technique was introduced in the context of image segmentation, the authors described how the method could be used as an optimization method for various functionals in [9]. In the present paper we show that the power watershed method of Couprie *et al.* is well-suited to address the surface reconstruction problem. The idea of using watersheds for surface reconstruction from a set of points is quite natural. It can be seen as a an extension of the classical "coffee bean" segmentation example, where a watershed is applied on a filtered distance function to separate overlapping convex objects [3].

For further comparison, we also propose in this paper to examine a well-known weighted isotropic discretization of (1) (for $q = 1$) known as "Total Variation" (TV). We mention that there exist other TV-based approaches developed to overcome metrication artefacts, *e.g.* [28,15]. However, due to space constraints, further comparison with these techniques will be the subject of future work. Finally, we will also show that our algorithm compares favorably with existing surface reconstruction algorithms [2,5,21,20,19].

2 Method

For solving the surface reconstruction problem, we first place the points cloud onto a regular grid. Our method aims to label the nodes of the grid as an indicator of the object to reconstruct. Since the power watershed is defined on a graph, we begin by casting the surface reconstruction problem formulation in discrete terms.

A graph consists of a pair $G = (V, E)$ with vertices $v \in V$ and edges $e \in E \subseteq V \times V$ with cardinalities $n = |V|$ and $m = |E|$. An edge, e, spanning two vertices, v_i and v_j , is denoted by e_{ij}. We define an edge set corresponding to a 4 or 8-connected lattice (or 6-, 18- or 26-connected in 3D). A weighted graph assigns a (typically non-negative and real) value to each edge called a weight. The weight of an edge e_{ij} is denoted by $w(e_{ij})$ or w_{ij}. A plateau is a maximal set of connected edges with identical weight.

Given foreground F and background B node values (also called seeds), and p, q two real positive values, the energy presented for binary segmentation in [8] is a discretization of (1) given by

$$\begin{aligned} &\min_x \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q \\ &\text{s.t.} \;\; x(F) = 1, \; x(B) = 0. \end{aligned} \tag{2}$$

In this energy, x is a labeling indicating the foreground and background membership. The edge values w_{ij} can be interpreted as weights enforcing a regularization of the contours, such that any (usually unwanted) high-frequency content is penalized in x. The definition of the weights for surface reconstruction from a set of points P is based on the construction of a discrete distance map d_P, in a grid bounding the set of points.

$$w_{ij} = \min\left(d_P(i), d_P(j)\right), \tag{3}$$

where $d_P(i)$ is the discrete Euclidean distance between the node i and the set of points P. We recall that exact Euclidean discrete distance map may be obtained in linear time using the algorithm of Hirata [17], and that high-quality ordered algorithms also exist [25] . The background seeds may simply correspond to the frame, or bounding box of the lattice. The foreground seeds can be given by the maxima of the distance function that are not connected to the frame. The distance map may be previously filtered, for example using an attribute filter to obtain more robust markers [6,24].

As we illustrate in the remainder, the energy defined in (2) essentially forces x to remain smooth within the object, while allowing it to vary quickly close to point clusters near the boundary of the object. The data constraints enforce fidelity of x to a specified configuration, taking the values zero and one as the reconstructed object indicator. Observe that the values of x may not necessarily be binary when the value of q is strictly greater than one, which is a positive point for the surface reconstruction problem as we will further explain in this work.

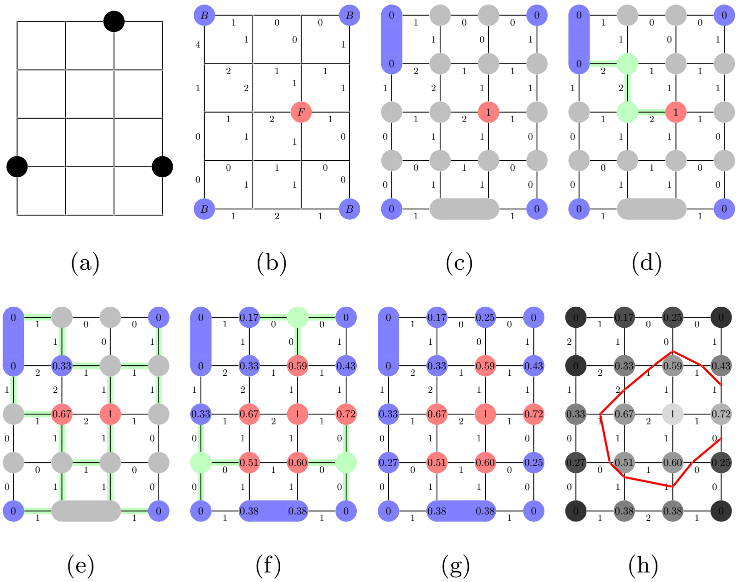
The different values of p and q lead to different algorithms for optimizing the energy. When the power of the weight, p, is finite, and the exponent $q = 1$, we recover the Graph cuts energy which can be optimized by a max flow algorithm. When $q = 2$, we obtain a combinatorial Dirichlet problem also known as the Random walker problem [16]. As described in [8,10], when the exponent p tends toward infinity, the cut obtained when minimizing the energy is a watershed cut [11], which has been shown to be equivalent to Maximum Spanning Forests [11] (MSF). Furthermore, an algorithm is presented to optimize the unique watershed that optimizes the energy for $q = 2$ and $p \to \infty$. The power watershed (PW) algorithm is recalled in Algorithm 1.

Algorithm 1. Power watersheds algorithm $p \to \infty, q = 2$

```
Data: A weighted graph $G(V, E)$ comprising known labels $x(B), x(F)$.
Result: A labeling $x$ solution of (2).
while any node has an unknown label do
    Find a maximal subgraph $S \in G$ composed of edges of maximal
    weight;
    if $S$ contains any nodes with known $x$ then
        Find $x_S$ minimizing (2) for $q = 2$ on the subset $S$;
        Consider all $x_S$ values produced by this operation as known;
    else
        Merge all of the nodes in $S$ into a single node, such that when the
        value of $x$ for this merged node becomes known, all merged nodes
        are assigned the same value of $x$ and considered known;
```

This set of parameters $q = 2$ and $p \to \infty$ is particularly interesting:

1. The power watershed algorithm has a well-defined behavior in the absence or lack of weight information (presence of plateaus). An example is shown at Figure (1).
2. The worst-case complexity of the power watershed algorithm in the case $p \to \infty$ is given by the cost of optimizing (2) for the given q. In the best-case scenario (all weights have unique values), the power watershed algorithm has the same asymptotic complexity as the algorithm used for a MSF computation (quasi-linear) (See [7] for more details). In practical applications where the plateaus have a size less than some fixed value K, then the complexity of the power watershed algorithm matches the quasi-linear complexity of the standard watershed algorithm.


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

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

Fig. 1. (a) Three dots in a 4 × 5 lattice. (b) Associated lattice weighted by a distance map from the dots according to (3) and squared, with Foreground and Background seeds. Note that only the ordering of the weights counts in the power watershed algorithm, thus the squared distance map can be used directly and produces the same solution as if the weights in (3) were used. (c) First steps of the power watershed algorithm to optimize (2) in the case $q = 2$ and $p \to \infty$. Nodes having a maximum weight are merged. (d) A plateau of weight 2 (in green) including different seeded nodes is encountered. The Random walker algorithm is applied to label the nodes on the plateau. (e,f) New plateaus of weight 1 and 0 are encountered, the Random walker algorithm is applied, (g) Final labeling x solution of (2). The isocontour is represented in red.

Although the PW algorithm is fast, it can be further accelerated for the specific case of weight defined according to a distance map (3). Given the foreground and background seeds, we define a narrow band $\mathcal{S}$ as the set obtained by thresholding the distance map d_P with the the smallest threshold T_S such that the connected components are divided between at least an interior (foreground) and an exterior (background). We then compute the PW only on this incomplete distance map, which saves both time and memory. In practice, it is possible to avoid computing an exact distance map on the full grid, using for instance an ordered algorithm propagating from the point cloud [25].

Recall that the exterior seed is connected to the frame of the image, so this is a simple unambiguous connectivity criterion. Applying PW on $\mathcal{S}$ is guaranteed to provide the same (unique) global optimizer of the energy as the solution on the full grid, because the connectivity criterion ensures this computation yields a Jordan curve (2D) or surface (3D) separating foreground and background seeds and passing through only already-computed distance values. Performing the PW computation on the full map would not change this result because no distance weight would be lowered, and so no new surface with a lower weight could be found on this full map. Conversely, thresholding the distance map at a lower value than T_S *would* change the result, as any Jordan curve or surface separating foreground and background seeds and computed on this map would necessarily cross some nodes where the distance map had not been computed. Note this does not mean that this thresholding criterion is necessary and sufficient for optimal computation and least memory usage, as an adaptive threshold depending on the local point density could be used instead. However, we leave it as future work to find a better criterion than the simple global threshold.

3 Results and Comparative Evaluation

3.1 Comparison with Graph Cuts and Total Variation

We now demonstrate the performance of the power watershed algorithm for surface reconstruction with respect to two graph-based methods discretizing the energy defined in equation (1), namely the weighted total variation (TV) and the Graph cuts (GC) method.

Our first experiment consists of finding a contour fitting sparse and noisy dots in a 2D plane. Figure 2 compares the result of TV minimization, the Graph cuts result, a maximum spanning forest result, and the result obtained with power watershed algorithm ($q = 2$, $p \to \infty$). We observe that all resulting contours are excluding outliers. The Graph cuts results demonstrate that this algorithm is less sensitive to noise, but the contours are blocky because the obtained object indicators are binary. We note that a post-processing step for producing a smooth isosurface from such a binary object reconstruction has recently been proposed by Lempitzky [22]. The TV contour is the smoothest one, which may be an unwanted effect for rendering details in surfaces. Thus, the

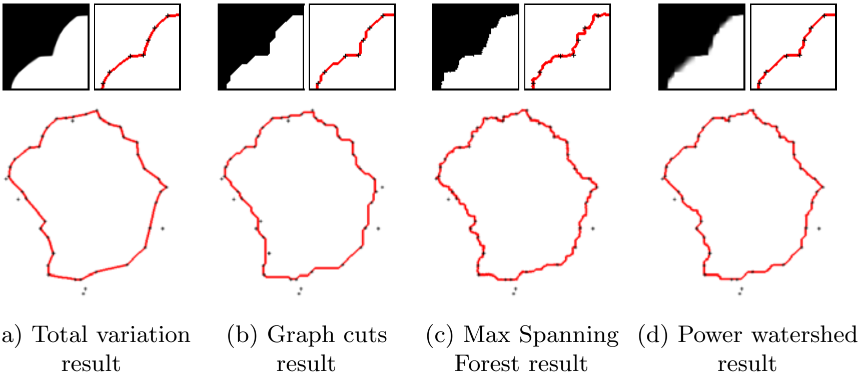
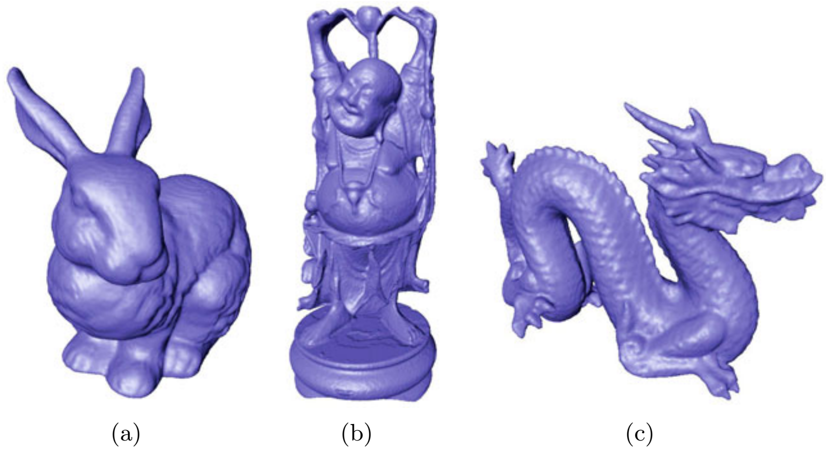
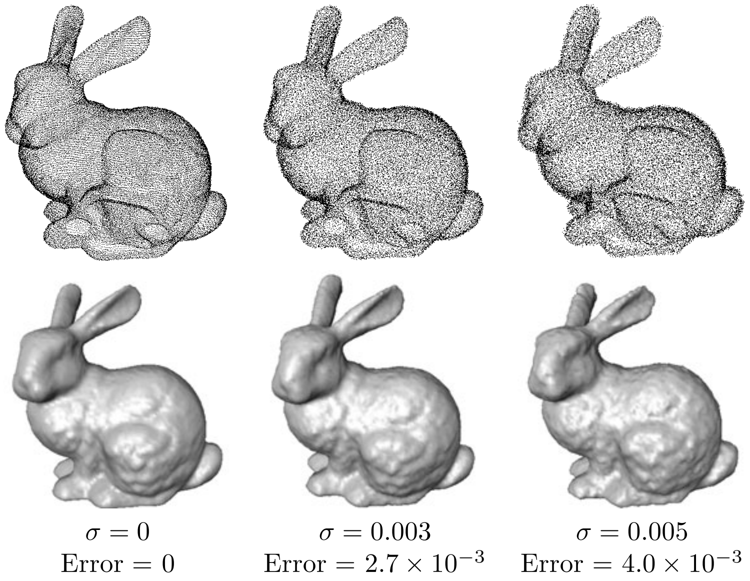

(a) Total variation result (b) Graph cuts result (c) Max Spanning Forest result (d) Power watershed result

Fig. 2. Comparison of surface reconstruction from a set of points in 2D, using the total variation method, Graph cuts, a maximum spanning forest algorithm and finally the power watershed algorithm



(a) (b) (c)

Fig. 3. Power watershed results obtained from points clouds. (a) Total size of scans : 362272 points, Grid size : 234×297×301. (b) Number of scans used : 341072 points, Grid size : 275 × 276 × 668 (c) Total size of scans : 2748318 points, Grid size 382 × 270 × 171.

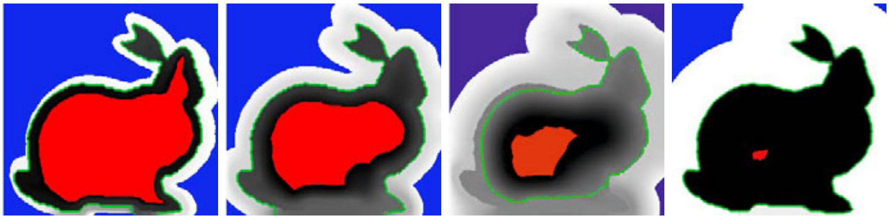


$\sigma = 0$ Error $= 0$

$\sigma = 0.003$ Error $= 2.7 \times 10^{-3}$

$\sigma = 0.005$ Error $= 4.0 \times 10^{-3}$

Fig. 4. Power watershed results obtained from noisy points clouds, corrupted by Gaussian noise of variance σ. The error was computed as the average distance between the obtained isosurface points to the original point cloud. The error is given in percentages of the diagonal of the bounding box of the data points.



(a) TV (122 s) (b) TV (301 s) (c) TV (950 s) (d) PW (32 s)

Fig. 5. Robustness to seed quantity: this figure shows slices of solutions obtained for reconstructing the bunny surface, obtained on a grid of size 251 × 248 × 195, using different seeding strategies. The interior seed is colored in red, and the background seed in blue, and resulting isolines in green. (a,b,c) : Results obtained with TV minimization, and (d) Result obtained with Power Watershed.

TV method requires us to adapt the parameter p in the exponent of the weights to the desired smoothness of the surface. The maximum spanning forest result passes through most points and the contour looks noisy. The power watershed result demonstrates good performance for fitting the dots, while avoiding both blocky contours and noise. This good performance is due to the presence of interwoven plateaus in the distance map. During the execution of the algorithm, the Random walker (RW) algorithm is called several times around the dots, resulting in a smooth output x. An isocontour or isosurface computation at the 0.5 level is thus providing smooth contours compared to the binary results obtained with Graph cuts or maximum spanning forests.

Figure 3 shows surfaces reconstructed from noisy scanned dot sets using the power watershed (PW) algorithm. We used the coordinates of points acquired from scans of several 3D shapes (bunny, Buddha) from the Stanford database available online [1] In our experiments, we embedded those points in 6-connected grids. Quantitative comparisons for the fitting quality are difficult because not all data points are required to be part of the surface. However, we show in Figure 4 that the power watershed method is producing reasonable results even if the point cloud is corrupted by Gaussian noise. We also compared our results to the results obtained using TV minimization and Graph cuts at Figure 6. We can observe that the surface obtained with the Graph cuts method is quite blocky. Using the same rendering method to render the output x minimizing the power watershed energy, the power watershed algorithm obtains a smoother surface showing significantly more details. Figure 5 shows that the power watershed method performs well even with a small amount of seeds. In contrast, the total variation method requires a large amount of seeds placed close to the searched surface.

3.2 Computation Times and Memory Requirements

In our comparisons, we used the C++ software library of Lemptitzky and Boykov available online which implements the touch-expand algorithm that minimizes

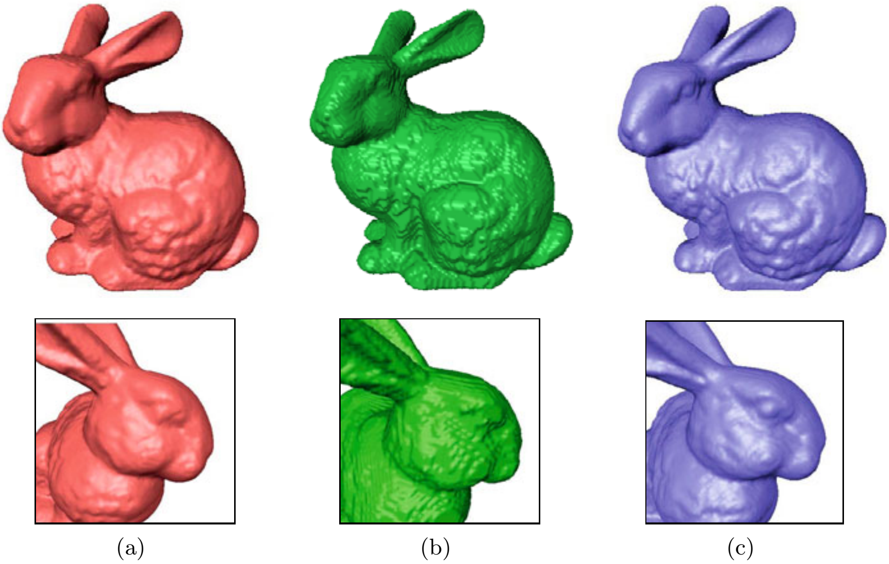


Fig. 6. Grid size : 234 × 297 × 301. (a) Total variation minimization result, (b) Graph cuts result, (c) power watershed result. Isosurfaces at 0.5 have been extracted on all results, and were downsampled by 2 to render the surfaces.

the GC energy. For that purpose, the touch-expand algorithm calls a max flow algorithm on a partial graph, a band around the points which is extended when the solution touches the boundary of the band. We also compare to a fast TV-based solver often called split-Bregman algorithm [14] but identical in principle to the Alternating Direction Method of Multipliers [13,26]. This TV solver is implemented in C and may be called from Matlab. Finally, we also implemented the PW method in C. Computation times of the compared algorithms for the bunny and Buddha data sets are provided on Table 1. The three methods have large memory requirements when applied to the full grid. However, the graph cuts and power watershed banded methods are less memory intensive. For the bunny dataset, the touch-expand algorithm optimizing the Graph cuts energy only needs to allocate 3.6% of the full grid size. The solution space for the bunny dataset is much larger for our power watershed algorithm and reaches 33% of the full grid, because some points are widely spaced out. Thus, on this image, the touch expand algorithm is faster than the power watershed method. However, points clouds may require a more extensive expansion of the research area for the Graph cut touch expand method. For example, in the Buddha dataset, 8.75% of the grid size is needed for computing the Graph cuts solution. In such cases, the power watershed approach is much faster. The TV approach has currently no guarantee to produce a global optimum when called on a banded graph, so the TV algorithm was implemented only on a full grid. On the Buddha dataset for example, given a 207 × 505 × 207 grid, the TV-based method requires 3G of RAM.

Table 1. Timing experiments (in seconds) using an PC with a 3GHz Intel dual-core processor and 2G of RAM. The memory requirements are given in Mega Octets, and the computation times in seconds. Note that the seeds used are different for the three methods. For TV and PW, the seeds are imposed as hard constraints and are located far from the point cloud (see Fig. 5). The GC touch expand method uses soft constraints seeds computed from the normals given in input with the point cloud. We observe that Graph cuts are fast when the point cloud configuration does not require to expand the research area too much, like in the Bunny case (3.6%). However in the Buddha example, there is a need for a larger expansion, where the real complexity of the Graph cut algorithm become visible.

Point cloud	Method	Grid size	Grid used	Peak memory	Time	Time / Nb voxels
Bunny	GC	$234 \times 257 \times 301$	3.6%	212	9	1.4×10^{-5}
Bunny	PW	$234 \times 257 \times 301$	31%	1180	51	9.1×10^{-6}
Bunny	TV	$195 \times 248 \times 251$	100%	900	122	1.0×10^{-5}
Buddha	GC	$276 \times 671 \times 277$	8.7%	850	253	5.6×10^{-5}
Buddha	PW	$276 \times 671 \times 277$	16%	1500	52	6.3×10^{-6}

3.3 Comparison with Some Other Approaches

Following the quantitative information given in [19], we compare the performances of power watershed for the Stanford bunny reconstruction with different methods [2,5,21,20,19]. All these methods, including the power watershed, reconstruct a surface close to the Bunny set of dots, with an error comprised between 2×10^{-4} and 6×10^{-4}. In terms of computation time, the Power Crust method [2] is about 10 times slower, and the Poisson method [20] 3 times slower than our PW approach. Although the FFT [21], MPU [5] and Hoppe *et al* [18] methods are fast, the FFT method suffers from large memory requirements limiting the grid resolution, and Hoppe *et al* and MPU methods produces artifacts in the presence of noise (See [19]). The method of Jalba and Roerdink [19], based on Coulomb potentials, uses 4 times less memory than our power watershed method, but is 5 to 10 times slower on a CPU and is still slower using a GPU. Furthermore our PW is more flexible in the choice of the markers. In our experiments, the amount of markers is not very large as shown in Fig. 5, but a strategy using larger seeds could be employed to reduce the size of the solution space and the computation time.

4 Conclusion

The power watershed method can be used to efficiently produce surfaces fitting noisy measurements. Contrary to standard watershed algorithms and the Graph cuts approach, the unique solution provided by the power watershed is not binary, resulting in the reconstruction of both smooth and detailed surfaces. In addition, this method is fast and not limited by large memory requirements, when using a restriction of the solution space. Finally, in comparison with other methods, our power watershed is robust to seed placement, and requires fewer parameters to be set. In practice, when using the power watershed method, close-fitting markers are not as mandatory as in other methods. Future work will follow several directions: memory and computation times improvements are still achievable, in particular by improving the touch-expand idea used in the Graph cuts optimization and adapting it for power watershed. The power watershed energy could also be modified to add some priors to the reconstructed surface, such as local orientation. Finally, we hope to demonstrate the efficiency of the power watershed technique for solving related problems such as multiview reconstruction.

References

1. Stanford 3D scanning repository, http://graphics.stanford.edu/data/3Dscanrep/
2. Amenta, N., Choi, S., Kolluri, R.K.: The power crust. In: Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, SMA 2001, pp. 249–266. ACM, New York (2001)
3. Beucher, S., Gratin, C.: Micromorph reference manual, applications and solutions. Ecole des Mines de Paris (1989)
4. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(11), 1222–1239 (2001)
5. Braude, I., Marker, J., Museth, K., Nissanov, J., Breen, D.: Contour-based surface reconstruction using mpu implicit models. Graphical Models 69 (2007)
6. Breen, E., Jones, R.: Attribute openings, thinnings and granulometries. Graphical Models and Image Processing Journal 64(3), 377–389 (1996)
7. Chazelle, B.: A minimum spanning tree algorithm with inverse-ackermann type complexity. J. ACM 47, 1028–1047 (2000)
8. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watersheds: a new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In: ICCV 2009, pp. 731–738 (2009)
9. Couprie, C., Grady, L., Najman, L., Talbot, H.: Anisotropic Diffusion Using Power Watersheds. In: ICIP 2010, pp. 4153–4156 (2010)
10. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power Watersheds: A Unifying Graph Based Optimization Framework. IEEE Transactions on Pattern Analysis and Machine Intelligence (to appear, 2011)
11. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(8), 1362–1374 (2009)
12. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Canadian Journal of Mathematics 8, 399–404 (1956)
13. Glowinski, R., Tallec, P.: Augmented Lagrangian and operator-splitting methods in nonlinear mechanics. SIAM, Philadelphia (1989)
14. Goldstein, T., Osher, S.: The split Bregman method for ℓ_1-regularized problems. SIIMS 2(2), 323–343 (2009)
15. Goldstein, T., Bresson, X., Osher, S.: Geometric applications of the split Bregman method: Segmentation and surface reconstruction (2009)

16. Grady, L.: Random walks for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11), 1768–1783 (2006)
17. Hirata, T.: A unified linear-time algorithm for computing distance maps. Information Processing Letters 58(3), 129–133 (1996)
18. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. SIGGRAPH Comput. Graph. 26, 71–78 (1992)
19. Jalba, A.C., Roerdink, J.B.T.M.: Efficient surface reconstruction using generalized coulomb potentials. IEEE Transactions on Visualization and Computer Graphics 13, 1512–1519 (2007)
20. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, SGP 2006, pp. 61–70. Eurographics Association, Aire-la-Ville (2006)
21. Kazhdan, M.M.: Reconstruction of solid models from oriented point sets. In: Symposium on Geometry Processing, pp. 73–82 (2005)
22. Lempitsky, V.: Surface extraction from binary volumes with higher-order smoothness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2010)
23. Lempitsky, V., Boykov, Y.: Global Optimization for Shape Fitting. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, USA (2007)
24. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Transactions on Image Processing 15(11), 3531–3539 (2006)
25. Ragnemalm, I.: The euclidean distance transform in arbitrary dimensions 14(11), 883–888 (1993)
26. Setzer, S.: Split Bregman algorithm, douglas-rachford splitting and frame shrinkage. In: International Conference on Scale Space and Variational Methods in Computer Vision, SSVM 2009, pp. 464–476. Springer, Heidelberg (2009)
27. Strang, G.: Maximum flows through a domain. Math. Prog. (26), 123–143 (1983)
28. Ye, J., Bresson, X., Goldstein, T., Osher, S.: A fast variational method for surface reconstruction from sets of scattered points (2010) (submitted)
29. Zhao, H.K., Osher, S., Merriman, B., Kang, M.: Implicit, nonparametric shape reconstruction from unorganized points using a variational level set method. Computer Vision and Image Understanding 80, 295–319 (1998)

Voxel-Based Assessment of Printability of 3D Shapes

Alexandru Telea[1] and Andrei Jalba[2]

[1] Institute Johann Bernoulli,
University of Groningen, The Netherlands
a.c.telea@rug.nl
[2] Department of Mathematics and Computer Science,
Eindhoven University of Technology, The Netherlands
a.c.jalba@tue.nl

Abstract. Printability, the capability of a 3D printer to closely reproduce a 3D model, is a complex decision involving several geometrical attributes like local thickness, shape of the thin regions and their surroundings, and topology with respect to thin regions. We present a method for assessment of 3D shape printability which efficiently and effectively computes such attributes. Our method uses a simple and efficient voxel-based representation and associated computations. Using tools from multi-scale morphology and geodesic analysis, we propose several new metrics for various printability problems. We illustrate our method with results taken from a real-life application.

Keywords: Geodesic analysis, Top-hat transform, 3D printability.

1 Introduction

Recent advances in 3D printing technology have made it possible to create faithful replicas of 3D models with increasingly higher resolutions, wider ranges of materials, higher printing speeds, and lower costs. High-quality, low-cost 3D printing is now available for personal users, *e.g.* through online services [12].

However, these advances have also created new challenges. Printer resolution limitations introduce several differences between input models and output objects, *e.g.*, in order of gravity: small layers of the surface being peeled, thin shape parts being fully removed, and shape break-up in several parts due to narrow connections. Currently, such defects are detected largely by manual inspection. This is not scalable for online printing services faced with thousands of models uploaded per day. What is needed is an automatic printability assessment method.

In this paper, we present a framework for 3D printability assessment. We analyze known defects which occur during printing and propose several metrics to detect and measure their criticality. We next compute these metrics fully automatically, but also allow users to tune parameters in order to select what is critical when printing a given model on a given printer. As 3D printing technology

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 393–404, 2011.

works in a raster fashion, we implement our proposed metrics using a voxel-based approach relying on multi-scale morphology and geodesic analysis. For maximal performance, we use a GPU implementation based on NVidia's CUDA.

The structure of this paper is as follows. In Section 2, we overview 3D printing basics and related work in shape thickness measurement. Section 3 presents the several stages of our printability assessment framework and related metrics. Section 4 presents results obtained on 3D models from a real-life 3D printing process. Section 5 discusses our method. Section 6 concludes the paper with future work directions.

2 Related Work

3D printing technology constructs a model layer by layer, in a process similar to voxelization. The main critical element here is the shape's local *thickness* t: Shape details thinner than the so-called printer resolution τ result in empty voxels, which can lead to the problems named in Sec. 3. Hence, local thickness detection is a necessary (but as we shall see, not sufficient) part of printability assessment.

Several methods address thickness measurement. Hildebrand *et al.* define local thickness t at a point $\mathbf{p}$ in a 3D shape S, given by a set of voxels $\Omega \subseteq \mathbb{R}^3$, as the diameter of the largest inscribed ball B centered at $\mathbf{x}$ that contains $\mathbf{p}$ [3], *i.e.*,

$$t(p) = 2\max(\{r \,|\, \mathbf{p} \in B(\mathbf{x}, r) \subset \Omega, \mathbf{x} \in \Omega\}), \tag{1}$$

with r the radius of ball B. Dougherty *et al.* implement the above by evaluating the distance transform $D(S)$ over the skeleton (or medial axis) of S and marking all points $p \in \Omega$ with the maximum of $D|_\Omega$ [3]. However, this method requires a robust and exact medial axis transform, which is prohibitively expensive for large voxel models.

Yezzi *et al.* compute thickness between surface pairs by minimal-length surface-to-surface paths using a PDE-based approach [14]. While this method is relatively fast (12 seconds for a 256^3 voxel dataset), its extension to higher-genus models is not evident. An overview of voxel-based thickness estimation in medical applications is given in [11].

Mesh-based methods for thickness computation have been proposed by Lambourne *et al.* [5]. However, they typically require a clean mesh with no self-intersections, do not offer a volumetric object representation (which is essential for subsequent printability assessments), are more complex to implement, and are only shown to work well on relatively simple shapes. Many engineering tools measure the so-called *wall thickness* [4]. However, all such tools we are aware of work only for models consisting of few relatively large, simple, surfaces separated by clear edges.

Considerable work exists in the area of topology-accurate 3D shape digitization, *e.g.* [13]. However, our problem is different: given a digitization process not under our control (the 3D printer), and a specific set of quality criteria (mainly thickness-related), we must assess whether a given polygonal model fullfils these criteria under the given digitization.

3 Printability Assessment

Printability assessment supplements thickness estimation with additional constraints. A 3D shape is printable when the removal of its thin regions (which do not print) do not create *critical* topological or geometrical changes, as follows (see also Fig. 1):

- disappearance of salient detail, or *spikes*, *e.g.* the mouse's tail or whiskers;
- disconnection of large shape fragments by removal of so-called *bridges*, *e.g.*, the mouse's thin arms;
- creation of large *holes*, also called tunnels, *e.g.*, center of sandal's sole.

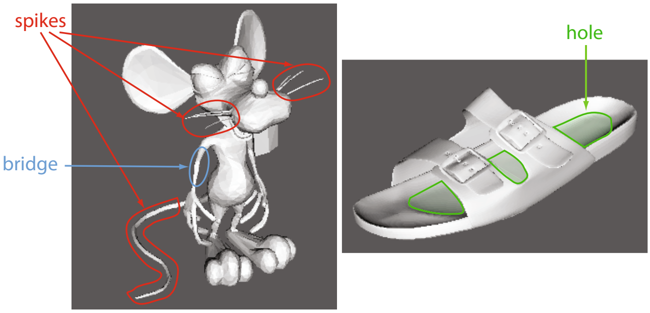


Fig. 1. Printability critical events: spikes, bridges, and holes (see also Fig. 5)

Filling of small object regions such as holes of interstices due to printer discretization is not seen as a problem in 3D printing practice, as such details are few in typical 3D shapes on the printing market. Also, filling does not cause parts of the object to break off or disappear, which is seen as the largest problem.

Following discussions with a market-leading 3D printing service provider [12], we learned that an effective printability assessment method should be able to detect and measure all above-mentioned events related to spikes, bridges, and holes for a shape of arbitrary topological and geometric complexity and given printer resolution. Secondly, the criticality of events should be quantifiable. Intuitively, this should match the visual salience of these events. Finally, the method should be automatic and efficient, *i.e.*, handle thousands of models a day on a PC computer. According to their experience, no such ready-to-use method exists.

We next present a method that efficiently and effectively implements the above requirements. Fig.2 shows our computational pipeline.

3.1 Voxelization

First, we transform the polygonal 3D input shape S into a binary voxel model. We use the method of Nooruddin and Turk [9], based on an optimized implementation of *binvox* [7], which delivers high performance and can handle any

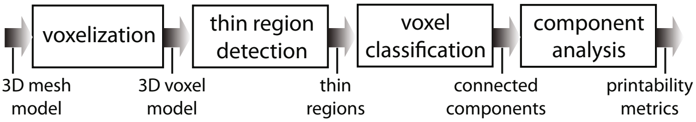


Fig. 2. Thickness assessment pipeline

closed mesh, including self-intersecting ones (see Sec. 4). The output is a binary volume consisting of the shape Ω (foreground) and its complement, the background, $\overline{\Omega} = \mathbb{R}^3 \setminus \Omega$. The voxelization resolution used in practice exceeds that of the 3D printer, so small potential errors caused by the voxelization methods should have no effect on our assessment.

3.2 Thin Region Detection

Given a user-defined thickness τ (equal to the printer resolution), we next detect the locally thin areas Θ of Ω. This can be formulated as a *top-hat* transform in the context of multi-scale morphology [1,6], as follows.

Given the set $\Omega \subseteq \mathbb{R}^3$, its distance transform with respect to the $||\cdot||_2$ (Euclidean) norm is defined as

$$D(\Omega)(\mathbf{x}) \equiv \bigwedge_{\mathbf{y}\in\overline{\Omega}} ||\mathbf{x}-\mathbf{y}||_2 \,. \tag{2}$$

Note that the distance transform of a shape can be computed both inside, respectively outside, of a particular shape, by inverting the roles of Ω and $\overline{\Omega}$ in Eq. 2, as inferred from the context of use.

The 3D unit ball B associated to the norm is

$$B \equiv \{\mathbf{x} \in \mathbb{R}^3 \mid ||\mathbf{x}||_2 \leq 1\}.$$

The *multi-scale set dilation* and *erosion* of Ω by B at scale s are given, respectively, by

$$\delta_B(\Omega, s) \equiv \Omega \oplus sB$$
$$\epsilon_B(\Omega, s) \equiv \Omega \ominus sB.$$

We detect locally thin areas Θ of the shape Ω using a multi-scale (set) top-hat transform at scale $s = \tau$, *i.e.*,

$$\Theta(\Omega, \tau) = \Omega \setminus \delta_B(\epsilon_B(\Omega, \tau), \tau). \tag{3}$$

The erosion in Eqn. (3) removes border elements, narrow spikes and bridges connecting the main object structures. Dilating the resulting set reconstructs the borders of the object, but not the other removed structures. Finally, taking the set difference between Ω and the smoothed (*opened*) object, the thin structures are effectively extracted. By setting τ to the printer resolution, we thus detect the unprintable parts, *e.g.* the blue areas in Fig. 3.

Since thresholding the distance transform $D(\Omega)$ at levels $s > 0$ yields multi-scale dilations of Ω by balls sB of radius s, *i.e.*,

$$\delta_B(\Omega, s) = \{\mathbf{x} \in \mathbb{R}^3 \,|\, D(\Omega)(\mathbf{x}) \leq s\} \tag{4}$$

and similarly,

$$\epsilon_B(\Omega, s) = \{\mathbf{x} \in \mathbb{R}^3 \,|\, D(\overline{\Omega})(\mathbf{x}) \geq s\}, \tag{5}$$

one can express Θ from (3) using (4),(5) as

$$\Theta(\Omega, \tau) = \Omega \setminus T_l(D(T_h(D(\overline{\Omega}), \tau)), \tau) \tag{6}$$

where T_h and T_l are the sets obtained by thresholding at level τ its (function) argument and by keeping only values higher and lower than τ, respectively.

The dilation replaces corners and edges from the eroded shape with spherical caps, respectively edge-swept cylindrical surfaces of radius τ, a well-known result from mathematical morphology. Practical observation shows that this rounding closely follows the actual behavior of 3D printers due to small deviations of the printer head of the order of the machine resolution.

Unlike other approaches (see Sec. 2), our thin-area estimator(Eq. (3)) can be very efficiently computed due to the result in Eq. (6), using a fast and exact method for computing the Euclidean distance transform (see Sec. 5). As an example, Fig. 5 e shows the volume-rendered distance transform of the rabbit model - blue indicates small distances, while red indicates large distances from the model's surface.

3.3 Voxel Classification

As already outlined, not all thin regions are equally critical for printability. Equation 3 does not distinguish between topological or geometric shape properties besides local thickness. For example, sharp edges would be classified as thin, while their removal amounts to limited rounding (Sec. 3.2), which is acceptable. Area or volume-based aggregated metrics are not sufficient for discrimination, *i.e.*, a shape with significant small-scale noise would yield thin regions with cumulative large area and/or volume, the removal of which would not significantly alter the shape. What is needed, is the detection and analysis of salient spikes, holes, and bridges.

To detect and analyze such events, we next classify the voxels into four categories (Fig. 3): thin (T), rump (R), interface (I) and boundary (B), as follows[1]:

$$\begin{aligned}
R &= \Omega \setminus \Theta \\
I &= \{\mathbf{x} \in \Theta \,|\, \exists \mathbf{y} \in n_6(\mathbf{x}), \mathbf{y} \in \Omega, \mathbf{y} \notin \Theta\} \\
B &= \{\mathbf{x} \in \Theta \,|\, \exists \mathbf{y} \in n_6(\mathbf{x}), \mathbf{y} \in \overline{\Omega}\} \\
T &= \Theta \setminus (B \cup I),
\end{aligned}$$

[1] We recommend viewing the figures in this paper in full color.

where $n_6(x)$ denotes the 6-connected voxel neighbors of $\mathbf{x}$. In a post-processing step, we next mark all I voxels with no B or T neighbors as R. This removes one-voxel-thick 'interface shells', *i.e.* thin surface-like voxel sets, which are tangent to rump, but not to thin regions, and thus have no further use in the printability assessment. Finally, we compute the thin (T) 26-connected components. For each component C_i, we store its I, T, and B voxels, boundary area A_i^B, and volume V_i. Also, for each C_i, we compute and store the 26-connected interface components I_{ij} as well as their areas A_{ij}^I. For robust area estimation on voxel surfaces, we use the fast estimator proposed by Mullikin and Verbeek [8].

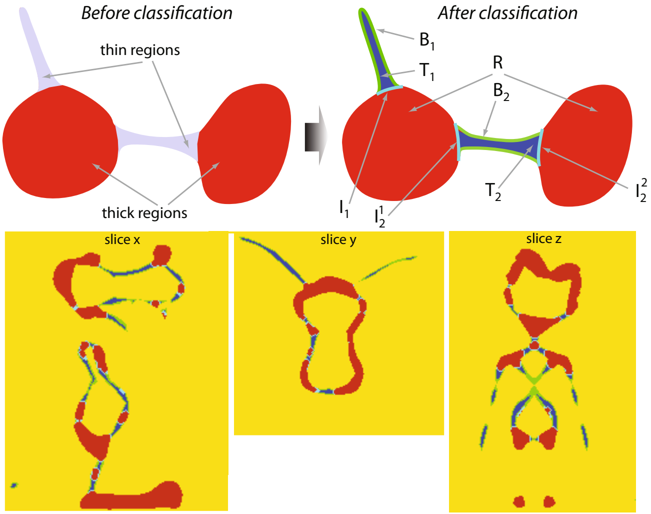


Fig. 3. Voxel classification into thin (T=blue), rump (R=red), interface (I=cyan), and boundary(B=green). Bottom: slices from actual voxel model in Fig. 1 left).

For example, the shape in Fig. 3 (top) has two thin regions, a spike and a bridge; the spike has one interface, while the bridge has two. Thin region classification is done by topological analysis: Bridges have several interfaces, $|\{I_{ij}\}_j| > 1$; spikes have an interface of genus zero; and holes have an interface of genus larger than zero. This information is important in assessing the printability metrics presented next.

3.4 Printability Metrics

We now describe two metrics which are computed on the thin components resulting from the voxel classification (Sec. 3.3) to assess a shape's printability.

Area-based metric. Spikes, bridges, and holes share one common property: they are thin, elongated, structures, which connect to the rump by small-size interfaces. The *visual salience* of such a structure C_i can be encoded as the ratio of their (total) boundary area $A_i^B = \sum_j A_{ij}^B$ and interface areas $A_i^I = \sum_j A_{ij}^I$, *i.e.*, $\mu_A(C_i) = A_i^B/A_i^I$. This metric has several desirable properties. First, it marks large removed structures with small interfaces (connections to rump), *e.g.* the ears of the mouse in Fig. 1, as more critical. This matches practical 3D printing insight: large structures are visually more salient, and the smaller their interfaces are, the higher the chance is that a 3D printer will not manage to render such connections, or that such connections will be mechanically fragile – an important property for the lifetime of printed shapes. Secondly, it treats spikes, holes, and bridges uniformly, without the need of a full-shape connectivity analysis. Computing μ_A from the classification results (Sec. 3.3) is straightforward.

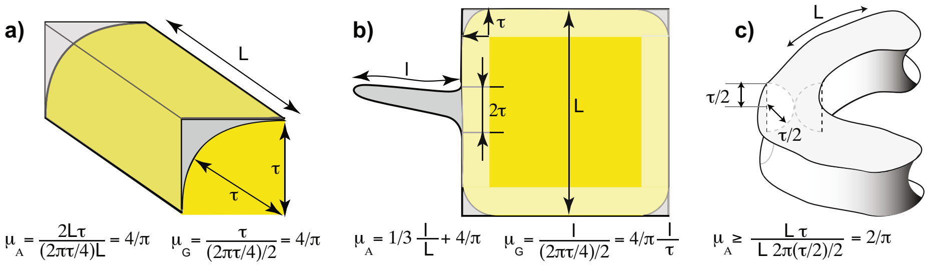

a)
L
τ
τ
$\mu_A = \frac{2L\tau}{(2\pi\tau/4)L} = 4/\pi$
$\mu_G = \frac{\tau}{(2\pi\tau/4)/2} = 4/\pi$
b)
τ
l
2τ
L
$\mu_A = 1/3\frac{l}{L} + 4/\pi$
$\mu_G = \frac{l}{(2\pi\tau/4)/2} = 4/\pi\frac{l}{\tau}$
c)
L
τ/2
τ/2
$\mu_A \geq \frac{L\,\tau}{L\,2\pi(\tau/2)/2} = 2/\pi$

Fig. 4. Printability metrics: edge (a); 2D view of spike on edge (b); thin hole (c); Thin areas are gray and rump areas are yellow respectively

The adimensional area-based metric μ_A is intuitive to interpret. For *spikes*, since interfaces are locally spherical (Sec. 3.2), μ_A is lower bounded by 1, when round shape parts are considered asymptotically thin for $\tau \to 0$. For right-angle edges, $\mu_A = 4/\pi$, *i.e.* the ratio of the area of two rectangles sharing the edge and having one side of length τ and the area of a quarter-cylinder (Fig. 4 a). Sharper spikes, which we want to detect, yield larger μ_A values. Near-constant cross-section spikes of length l, *e.g.*, the mouse's tail or whiskers in Fig. 1 have μ_A of the order of l/τ, since A^I is of the order of τ^2. This allows one to easily threshold μ_A to get all thin, elongated features longer than a given l. For *bridges*, consider a bridge component C_i having n branches $j = 1 \ldots n$ of boundary and interface areas A_{ij}^B and A_{ij}^I, respectively, so $\mu_A = \sum_j w_j(A_{ij}^B/A_{ij}^I)$, where $w_j = A_{ij}^I/A_i^I$. Hence, a bridge's metric is the sum of its branches' area metrics weighted by their interface sizes. For *holes*, μ_A is lower bounded by $2/\pi$, *i.e.* the ratio of a surface component of constant thickness 2τ and perimeter L and a cylinder surface of radius τ and length L (Fig. 4 c).

Geodesic length metric. However effective, the area-based metric will fail detecting certain complex unprintable structures. Imagine a thin cylindrical filament of radius τ and length l touching the edge of a cube of size L (Fig. 4 b). The thin region T (gray in Fig. 4 b) contains the filament *and* all edges of the cube, hence,

$$\mu_A = \frac{2\pi\tau l + 12(2\tau L)}{12(2\pi\tau/4)L} = \frac{1}{3}\frac{l}{L} + \frac{4}{\pi}. \tag{7}$$

If L is much larger than l, $\mu_A \to 4/\pi$, the right-angle edge metric, so we miss the unprintable filament. Note that this only occurs if the filament and cube edges form a *single* component, otherwise the filament retains its separate, large, μ_A value.

To solve this problem, we introduce a different metric, μ_G, based on geodesic lengths. For a given component,

$$\mu_G = \frac{\max_{\mathbf{x}\in B,\mathbf{y}\in I} ||(\mathbf{x} \rightsquigarrow \mathbf{y})||_2}{\max_{\mathbf{x}\in \partial I,\mathbf{y}\in I} ||(\mathbf{x} \rightsquigarrow \mathbf{y})||_2}, \tag{8}$$

where $(\mathbf{x} \rightsquigarrow \mathbf{y}) \subset \partial\Omega$ is a geodesic *path* on the shape surface $\partial\Omega = B \cup I$ between points $\mathbf{x}$ and $\mathbf{y}$.

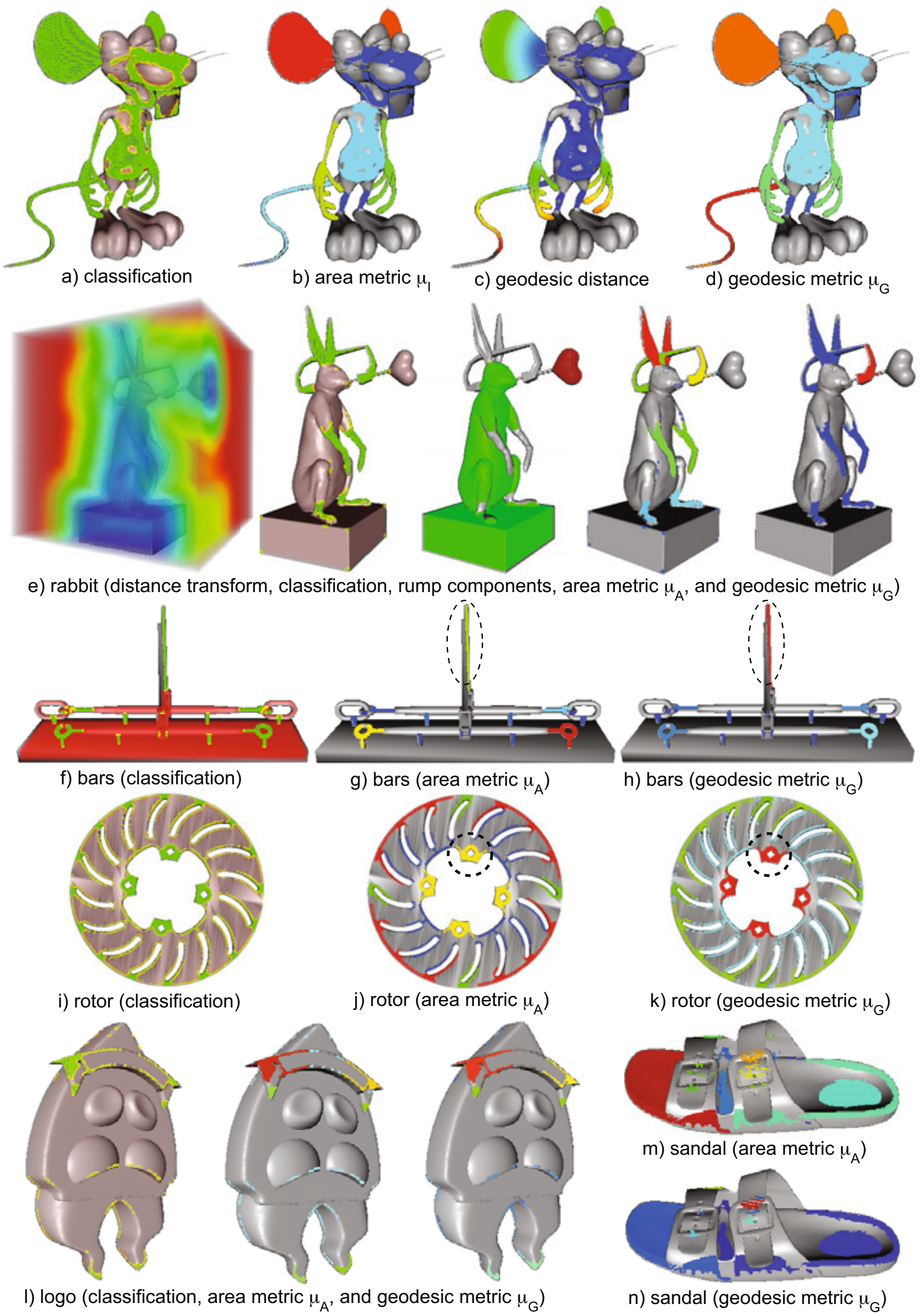
The adimensional metric μ_G measures the eccentricity of a component. For right-angle shapes, $\mu_G = \mu_A = 4/\pi$. Long *spikes* have high μ_A values. For the filament-and-cube shape, $\mu_G = (4/\pi)(l/\tau)$, *i.e.*, the value of μ_A for a right-angle-shape times the filament's eccentricity l/τ. For *bridges*, μ_G reflects half of the length of the longest interface-to-interface geodesic path, which is useful, as such branches are the likeliest to break after printing. For *holes*, μ_G equals the hole's surface diameter divided by the hole's wall thickness, which intuitively marks large-area, thin, regions as most critical.

We efficiently compute μ_G using the distance transforms $D(\partial I)|_B$ and $D(\partial I)|_I$ restricted to the voxels of B and I, respectively. These are nothing that the application of distance transforms in 3D voxel space on the voxel sets of B and I from the sites-set ∂I respectively. Note that, implementation-wise, ∂I is also included in B.

The maximum values of $D|_B$ and $D|_I$ give the maximal geodesic lengths from ∂I to all points in B and I, respectively. Since these distance transforms are computed on 3D surfaces rather than volumes, we cannot use volumetric distance transforms (see Sec. 5). Instead, we use the exact Euclidean TFT method [10], as it is simple to implement and can handle curved voxel surfaces. Any other fast, exact distance transform on curved voxel surfaces can be equally used. The Euclidean TFT is $O(N \log N)$ for a surface of N voxels. For typical B and I surfaces in a 512^3 voxel volume, a CPU-based Euclidean TFT takes a few seconds. As an example, Fig. 5 c shows the geodesic distances for the thin fragments of the mouse model. The tail's tip (red) sticks out as being the furthest point from the body.

4 Results

Figure 5 shows several results computed on a MacBook Pro laptop with 4 GB RAM and a GeForce 8600M GT card with 512 MB RAM. The models range from a few thousand to over hundred thousand polygons, and clearly contain non-trivial geometry and topology. All models are voxelized at 256^3 resolution.



a) classification

b) area metric μ_I

c) geodesic distance

d) geodesic metric μ_G

e) rabbit (distance transform, classification, rump components, area metric μ_A, and geodesic metric μ_G)

f) bars (classification)

g) bars (area metric μ_A)

h) bars (geodesic metric μ_G)

i) rotor (classification)

j) rotor (area metric μ_A)

k) rotor (geodesic metric μ_G)

m) sandal (area metric μ_A)

l) logo (classification, area metric μ_A, and geodesic metric μ_G)

n) sandal (geodesic metric μ_G)

Fig. 5. Printability assessment results on complex models in the 3D printing industry

The classification (using the same colormap as in Fig. 3) and and metric values (using a blue-to-red colormap) are rendered composited with the original polygonal surface (in gray). We see that the classification accurately detects thin regions, even in complex cases. For example, the mouse's torso is largely green, since the model is here actually not full, but consists of a thin shell – see Fig. 3 bottom. The area metric marks the ears as the most critical thin components. Indeed, these have a very large area but very small interfaces. However, compared to the ears, the tail is much less important, as it has a relatively small area compared to its interface. The geodesic metric also finds the ears as critical, but also finds the tail, which is thin and very elongated. A similar phenomenon occurs for the 'bars' model (Fig. 5 f-h). Here, the long vertical spike (marked with a stippled line) is clearly detected by the geodesic metric. For the rotor model (Fig. 5 i-k), the area metric identifies fragments of the rotor's outer edge (red in Fig. 5 j) as most critical. Note that these are bridges, and their removal would change the rotor's blade topology, *i.e.* disconnect its blades. The geodesic metric finds the four inner attachment parts (red Fig. 5 k) as most critical, as they are the most elongated thin regions to get removed. For the 'logo' model (Fig. 5 l), the two metrics identify critical fragments similarly. Finally, the sandal model shows how genus 0 and genus 1 thin regions (the sandal tip and heel, respectively) are captured, and that both the area and geodesic metric have values which do not depend on the thin region's genus, which desirable for user setting of the critical metric value.

The area metric captures critical fragments of large surface *area*, while the geodesic metric captures fragments of large *elongation*. The two are different, but equally critical, printability problems. In practice, we found that marking those fragments as printing problems, which are found critical by either metric, to be a robust and good solution to our overall assessment goal.

We validated the relevance of these results, *i.e.* the fact that the regions detected as critically thin for 3D printing by our algorithm would indeed create problems in actual 3D printing production in practice. Although the validation only involved domain experts looking at our results and not physical printing, the outcome was positive: our method identifies regions which a domain specialist would also mark as critical.

5 Discussion

Printability assessment. Our method is able to capture all top-level printability requirements (Sec. 3) by means of two simple, adimensional metrics. The metrics work uniformly for all types of events (spike and bridge removal and hole creation) on all geometric and topological configurations we could test on around 100 models used in 3D printing production. The entire method has only two parameters: the printer resolution τ and the desired, critical-event detection threshold for the used metrics. For now, the method can be used in a semi-automatic manner: the algorithm processes a set of shapes, identifies those with printability problems, and displays them (with highlighted problems as in Fig. 5) for further human inspection. Although not fully automatic, this is a major step forward as compared to the current procedure where users can only see the original 3D shapes.

Scalability. The most expensive step of our method, the thin region detection (Sec. 3.2), is implemented by adapting the recent CUDA-based distance transform (DT) of Cao *et al.* [2]. On our platform (Sec. 4), this step takes under 2 seconds for 256^3 voxels. Since Eqn. 6 uses only distance transforms and thresholding operations, we can safely extrapolate the timings from [2] for a Tesla C1060 card to our case, *i.e.* 0.7 seconds (512^3 voxels) and 5.8 seconds (1024^3 voxels). Since we work in voxel space, polygon count for the input model does not affect these values. Apart from thin region detection, all other steps are of negligible costs (seconds) and thus implemented on the CPU in C++.

Our current CUDA implementation requires 8 bytes GPU memory per voxel, *i.e.* 128 MB for 256^3 voxels, 1 GB for 512^3 voxels, and 8 GB for 1024^3 voxels. Current 1 GB cards thus allow measuring printability of objects of 10 cm size at the resolution of $100/512 \simeq 0.2$ mm, which is in line with the practical requirements for 3D printability assessment. It is, however, possible to improve on this by using the slice-based DT computation refinement sketched in [2], which allows volumes up to 8192^2 voxels per slice to be handled with only 512 MB with a small performance loss.

Precision. All distances we compute (CUDA DT for the thin regions and Euclidean TFT for the geodesic length metric) are exact. This is essential as small errors may assess a printable object as unprintable or conversely, which is unacceptable by the users.

Limitations. Voxelization resolution is currently the main limitation. Critical events smaller than one voxel will be missed. For example, the mouse's whiskers are not captured by the 256^3 voxel grid used in Fig. 5. However, as outlined above, this problem is directly solved by using graphics cards which can hold larger voxel volumes.

6 Conclusions

We have presented a solution for the task of automated assessment of 3D printability. For this, we combine a fast and robust method for thin region detection based on distance fields with two new application-specific metrics that uniformly treat all critical printability defects (spike and bridge removal and hole creation) and effectively capture critical events such as large surfaces having thin connections and elongated fragments. The entire pipeline requires only two user parameters: the printer resolution and maximal criticality of admitted defects, the latter which is an adimensional threshold value with intuitive border values. Our current implementation was tested in actual production runs at a 3D printing company and is currently forming the basis of an actual production tool for large-scale 3D printability assessment. In the future, we plan to extend our method in directly assessing 3D printability on mesh models, for optimal performance and accuracy, and also to design new metrics to capture more refined printability problems, such as specific topological and geometrical configurations in combination with specific material properties. Next, we plan to use our method in other application areas where shape thickness measurements are needed, such as 3D metrology for CAD/CAM and engineering.

Acknowledgements

We thank Robert Schouwenburg, CTO of Shapeways, for triggering this research, providing us with 3D printing details, and assessing our work, and Jacek Kustra (Philips Applied Technologies) for insightful technical discussions.

References

1. Brockett, R., Maragos, P.: Evolution equations for continuous-scale morphological filtering. IEEE Trans. Sig. Proc. 42(12), 3377–3386 (1994)
2. Cao, T., Tang, K., Mohamed, A., Tan, T.: Parallel banding algorithm to compute exact distance transform with the GPU. In: Proc. ACM SIGGRAPH Symp. on Interactive 3D Graphics and Games, pp. 134–141 (2010)
3. Dougherty, R., Kunzelmann, K.: Computing local thickness of 3D structures with ImageJ, 2007. In: Proc. Microscopy & Microanalysis Meeting, Ft. Lauderdale, Florida, August 5-9 (2009), www.optinav.com/LocalThicknessEd.pdf
4. Geometric Global. GeomCaliper tool (2010), geomcaliper.geometricglobal.com
5. Lambourne, J., Brujic, D., Djuric, Z., Ristic, M.: Calculation and visualisation of the thickness of 3D CAD models. In: Proc. SMI, pp. 107–112 (2005)
6. Maragos, P.: Differential morphology and image processing. IEEE Trans. Image Processing 5(6), 922–937 (1996)
7. Min, P.: Binvox voxelizer (2010), http://www.cs.princeton.edu/~min/binvox
8. Mullikin, J., Verbeek, P.: Surface area estimation of digitized planes. Bioimaging (1), 6–16 (1993)
9. Nooruddin, F., Turk, G.: Simplification and repair of polygonal models using volumetric techniques. IEEE TVCG 9(2), 191–205 (2003)
10. Reniers, D., Telea, A.: Tolerance-based feature transforms. In: Jorge, J., et al. (eds.) Advances in Comp. Graphics and Comp. Vision, pp. 187–200. Springer, Heidelberg (2007)
11. Sato, Y., Nakanishi, K., Tanaka, H., Nishii, T., Sugano, N., Nakamura, H., Ochi, T., Tamura, S.: Limits to the accuracy of 3D thickness measurement in magnetic resonance images. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, pp. 803–810. Springer, Heidelberg (2001)
12. Shapeways. Online 3D printing (2010), www.shapeways.com
13. Stelldinger, P., Terzic, K.: Digitization of non-regular shapes in arbitrary dimensions. Image and Vision Computing 26(10), 1338–1346 (2008)
14. Yezzi, A., Prince, J.: An Eulerian PDE approach for computing tissue thickness. IEEE Trans. Med. Imag. 22(10), 1332–1339 (2003)

A Comparison of Two Tree Representations for Data-Driven Volumetric Image Filtering

Andrei C. Jalba and Michel A. Westenberg

Department of Mathematics and Computer Science,
Eindhoven University of Technology,
Den Dolech 2, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{a.c.jalba,m.a.westenberg}@tue.nl

Abstract. We compare two tree-based, hierarchical representations of volumetric gray-scale images for data-driven image filtering. One representation is the max-tree, in which tree nodes represent connected components of all level sets of a data set. The other representation is the watershed tree, consisting of nodes representing nested, homogeneous image regions. Region attribute-based filtering is achieved by pruning the trees. Visualization is used to compare both the filtered images and trees. In our comparison, we also consider flexibility, intuitiveness, and extendability of both tree representations.

Keywords: attribute filtering, max-tree, watershed, waterfall tree.

1 Introduction

Hierarchical, region-based representations of scalar images have been thoroughly studied, see [1,17,18,19] and references therein. Most methods rely on a tree representation, whose nodes represent regions of aggregated pixels, at different representation scales. In the scalar case, the initial partition (of regions) from which the tree is built, is given by the set of flat zones [17] or regional maxima/minima [5,18] of the input image, or by an initial over-segmentation [1], through the morphological watershed transform [4,16].

Such image representations offer many advantages over traditional, pixel-based representations [5,17,18,19], e.g., they are much more efficient, as the number of regions is much smaller than the number of original pixels [17]. Manipulating such representations can be done very efficiently, so that interactive segmentation becomes possible [1,9,10,11].

In this paper, we compare two tree-based, hierarchical representations of 3D gray-scale images for data-driven image filtering. The first representation we consider is the so-called max-tree [18] data structure, in which tree nodes represent connected components of all level sets of a data set. The second representation is what we call the *watershed tree*, consisting of nodes representing nested, homogeneous image regions. *Image filtering* is achieved by first computing region *attributes* and comparing them with a user-provided threshold value. Then, those tree nodes representing regions for which the attribute value is smaller than the

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 405–416, 2011.

threshold are deleted, and the corresponding image regions are merged, so that effectively, image filtering or simplification is achieved. To support the filtering task, various shape and size region descriptors were implemented. The resulting filtered 3D images *and* trees are visualized for comparing flexibility, intuitiveness, and extendability of both tree representations

2 Background

Let $\mathbf{M} \subseteq \mathbb{R}^3$ be the volume domain, and $f : \mathbf{M} \rightarrow \mathbb{R}$ the gray scale volume. Implicitly, we assume the existence of some neighborhood graph on $\mathbf{M}$. Let $X \subseteq \mathbf{M}$ be a subset of $\mathbf{M}$. Set X is *connected* when for each pair (x_0, x_n) of points in X there exists a path of adjacent points in X that joins x_0 and x_n. A *connected component* of X is a connected set $C(X)$ which is maximal in size. A *flat zone* L_h at level h of f is a connected component $C(X_h(f))$ of the level set $X_h(f) = \{p \in \mathbf{M} \,|\, f(p) = h\}$. A *regional maximum* M_h at level h is a flat zone which has only strictly lower neighbors. A *peak component* P_h at level h is a connected component of the threshold set $T_h(f) = \{p \in \mathbf{M} \,|\, f(p) \geq h\}$.

2.1 Watershed Tree

A *watershed tree* is a hierarchical representation obtained by merging homogeneous regions of f, similar to the 'Live surface' method in [1]. However, unlike [1] we use the *waterfall* method [4] to group similar regions. Accordingly, our method starts with an initial partitioning of f into regions, computed by the watershed transform [4,16], applied to the gradient-magnitude image g of f. Then, regions are successively merged, based on a combined similarity and homogeneity criterion. The watershed tree itself encodes the nesting of the regions, due to the merging process. Thus, the *leaves* of the tree represent the initial partitioning of f. *Internal nodes* represent regions obtained by merging the regions corresponding to their children. The *root node* represents the entire image support $\mathbf{M}$. The tree-construction process is largely based on the fast, greedy algorithm in [8] for computing waterfalls.

The watershed tree represents a set of regions at different scales, and can be regarded as a hierarchical, region-based representation of f. Clearly, the tree does not encode all possibilities for merging regions belonging to the initial partition, but merely the most useful merging steps. Thus, both the *merging order* and the *region model* upon which the tree-construction process relies [17] have to be carefully chosen.

The region model of region $R_i \in \mathbf{M}$ is given in our method by the mean value $\overline{R_i}$ of f restricted to voxels in R_i. During the merging process, the mean value of the resulting region, corresponding to a parent node, is updated to reflect the union of the regions spanned by its child nodes. The merging order prioritizes regions which are both similar and homogeneous, and it is determined by *Student's* two-sample, T-square statistic

$$t^2 = \frac{N_i\, N_j\, \left(\overline{R_i} - \overline{R_j}\right)^2}{N_i\, \sigma_i^2 + N_j\, \sigma_j^2} \tag{1}$$

where $N_i = |R_i|$ and σ_i^2 is the variance of gray values in R_i. If the region model is updated during the merging process, the merging order is also recomputed.

2.2 Max-Tree Data Structure

A *max-tree* is a tree data structure where the nodes represent sets of flat zones of f. Specifically, a max-tree node C_h^k at level h, indexed by k, represents one or more flat zones at gray level h contained in a single peak component at the same level. Accordingly, the region model of a region R_i, corresponding to a flat zone L_h, is simply given by h. The merging order is given by the nesting of peak components at successive gray levels in f. Leaf nodes of the tree represent regional maxima of f, internal nodes represent flat zones at decreasing gray levels, whereas the root node represents the flat zones at the smallest gray level of f. Note that since a single max-tree node may represent multiple flat zones, it is impossible to distinguish among the corresponding image regions. Further, both the region model and merging order are fixed and dictated by the nesting of the peak components of f.

The dual tree structure is called a min-tree, and it can be obtained by constructing a max-tree of $-f$. The component-tree [5] is essentially the same as a max-tree (min-tree) in that it also represents the connected components of all level sets, yet in a more redundant way.

The max-tree can be constructed by a recursive flooding procedure that makes use of a hierarchical FIFO queue to process the voxels in correct order [18], or by relying on Tarjan's union-find algorithm [3,14].

2.3 Attributes

Once a tree is constructed, various shape or size *attributes* of the regions represented by the tree nodes can be computed. In principle, any attribute can be used. For computational reasons, however, it is advantageous to have attributes that can be computed *incrementally*, requiring only one traversal of the tree [21].

Volume: The most straightforward increasing attribute is the volume

$$V(R) = |R|, \tag{2}$$

which is simply the number of voxels of region R.

The following attributes are all based on the *moment-of-inertia tensor* $I(R)$ of region R, and its eigenvalues $\lambda_1(R)$, $\lambda_2(R)$, and $\lambda_3(R)$, assumed to be sorted in decreasing order of magnitude. Tensor I is essentially the covariance matrix of R multiplied by the number of voxels in R, see [21] for details of its computation and the derivation of the following attributes:

Non-compactness: This scale-invariant attribute has been proposed in [22] for vessel-enhancement filtering, and it is defined as

$$\mathcal{N}(R) = (\operatorname{Tr} I(R))/V^{5/3}(R). \tag{3}$$

Elongation: Intuitively, $\lambda_1(R)$ describes the length of the major axis of the ellipsoid representation of $I(R)$. Elongated regions will therefore have a high value of the ratio

$$\mathcal{E}(R) = |\lambda_1(R)/\lambda_2(R)|. \tag{4}$$

Flatness: Flat regions are characterized by a small value of $\lambda_3(R)$ in comparison to the other two eigenvalues. Such regions have a high value of the ratio

$$\mathcal{F}(R) = |\lambda_2(R)/\lambda_3(R)|. \tag{5}$$

Sparseness: A region is sparse when its expected volume, given its principal axes, is large in comparison to its actual value. This is expressed by

$$\mathcal{S}(R) = \pi d_1 d_2 d_3/(6V), \tag{6}$$

where $d_i(R)$ estimates the length of the principal axis corresponding to λ_i.

3 Filtering and Simplification

In general, there are two approaches for simplifying a tree representation [17,19]: (i) pruning and (ii) non-pruning strategies. Filtering is governed by a *criterion* involving shape or size attributes, such as defined above. Tree filtering is a pruning strategy, if the filtering criterion is *increasing* [19]. However, since most (shape) attributes are non-increasing, in general, filtering becomes a non-pruning strategy. Thus, in the latter case it makes sense to speak about filtering rules, see below and [17,19].

In the filtering step, the attribute of each tree node is assessed against the filtering criterion, *i.e.*, each node attribute's value is compared to a given threshold τ. If its attribute value is smaller than τ, the node does not meet the criterion and is labeled accordingly. Pruning of the tree branches can then be done with different, simple strategies, such as [18,20]

- The max rule: branches are pruned from leaves up to the first ancestor that has been labelled for preservation.
- The min rule: branches are pruned from leaves up to the last ancestor that has been labelled for removal.
- The direct rule: simply removes a node, and merges its members with the node of the first ancestor that has been labelled for preservation.
- The subtractive rule: as the direct rule, but lowers the gray levels of surviving descendants of removed nodes to preserve local background contrast.

Any of the filtering strategies above can be used to simplify a max-tree, and thus to filter the input image. For the watershed tree, only the first three strategies are meaningful. The subtractive rule implicitly assumes that an ordering of flat zones (i.e. by gray levels) is available, which is not the case for the watershed tree constructed from an initial partitioning of the gradient image g.

An additional means for image filtering through the watershed tree is what we call *level selection* – a pruning strategy which performs a cut at the same depth

τ_d, along each path from leaf to root. Accordingly, all nodes on the leaf-side of the cut are *collapsed* onto the last surviving ancestors along the root-side of the cut. Both the watershed-tree and the volume rendering views (see Section 4) are updated to reflect the cut. At any depth τ_d, level selection can be regarded as a region-based clustering method, using a simultaneous region similarity and homogeneity criterion. Note that in the max-tree case, level selection would simply become a (less useful) *thresholding* operation. Further, since (shape) attribute values may vary widely, as regions in the watershed tree merge, some stabilization is achieved by combining level selection with tree filtering. That is, first a depth d is selected, and then the *resulting*, pruned watershed-tree is filtered using the *most conservative* strategy, namely the max rule. Throughout this paper we use this combined strategy to filter the watershed tree.

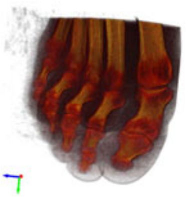
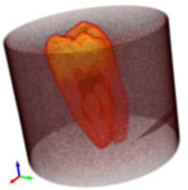
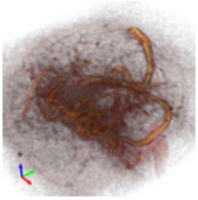
4 Visualization

We use *direct volume rendering* to visualize the 3D images. Conceptually, this technique works by casting a viewing ray perpendicular to the view plane from each pixel in the image into the volume. This ray is then sampled at regular intervals, and the samples are composited to yield a final color for the pixel. We implemented this rendering process by making use of texture mapping hardware [6] available on standard graphics cards.

Material properties, such as color and opacity, are assigned to voxels by a *transfer function* (TF). We have implemented a *standard* TF and propose a new *tree-based* TF. The standard TF simply maps the gray value corresponding to the region model of the given tree to a color value from a heated body color map, which ranges from black, via red and yellow, to white. The tree-based TF colors entire regions in the volume data, which are encoded in the tree by subtrees. At initialization, each tree node is assigned a random color. The attribute value is then inspected along the path from leaf to root: if the change in normalized attribute value between parent and child is above some threshold, the node is flagged. After processing all paths, the color values of the flagged nodes are propagated through the tree, i.e., a node is given the color of its parent unless it is flagged itself. For both TFs, the opacity is assigned from the gradient magnitude [7] to emphasize borders between regions.

The trees themselves are also visualized; the nodes of a tree are positioned by a radial layout algorithm [2]. The root node forms the center of the drawing, and the other nodes are arranged on concentric circles around the root. The depth of the tree determines the number of circles required to lay out the entire tree. Each subtree is allocated a sector of the circle proportional to its size, in which it is laid out. In the drawing, the size of a node is proportional (logarithmically) to the volume of the region associated with it, and the color of a nodes corresponds to the color assigned by the TF.

Table 1. Description of the data sets

	Foot		**Piggy bank**		**Tooth**		**Angiogram**	
	256 × 256 × 256		256 × 256 × 180		256 × 256 × 161		186 × 246 × 124	
								
	MT	WT	MT	WT	MT	WT	MT	WT
Nodes	279513	534083	19752	121189	133316	405612	332169	678977
Leaves	112223	327199	14038	75953	90501	246124	129670	410663
Depth	256	15	256	13	254	16	256	15
Branching	68.82	2.60	4.38	2.70	4.61	2.52	1.04	2.45

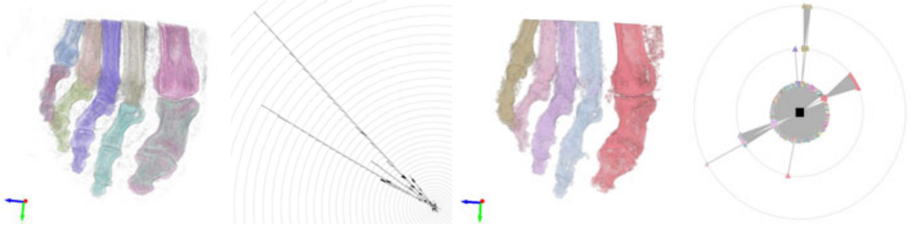
5 Data-Driven Comparison

We use a number of volumetric data sets available publicly[1]: *Foot*, *Piggy bank*, *Tooth*, all CT scans, and *Angio*, an MR angiogram. Table 1 shows dimensions and initial visualizations of these data sets.

A structural comparison of both trees in given in Table 1. We consider the total number of nodes, the number of leaf nodes, the depth, and the average branching factor. The branching factor per level is computed by N_l/N_{l+1}, where l denotes the level and N the number of nodes at that level. The average branching factor is then simply the average of the branching factors of all levels. As can be seen in the table, the number of internal nodes and leaf nodes in both trees is strongly data dependent: the noisy data sets have considerably more nodes than the Piggy bank. The max-tree has a depth that is always equal to the number of gray levels present in the input data. The watershed tree is considerably less deep, because of the different region model. The average branching factors show that the watershed tree has a more or less constant branching factor. This is in contrast to the max-tree, for which these numbers vary substantially, suggesting that a max-tree is much less regular than a watershed tree.

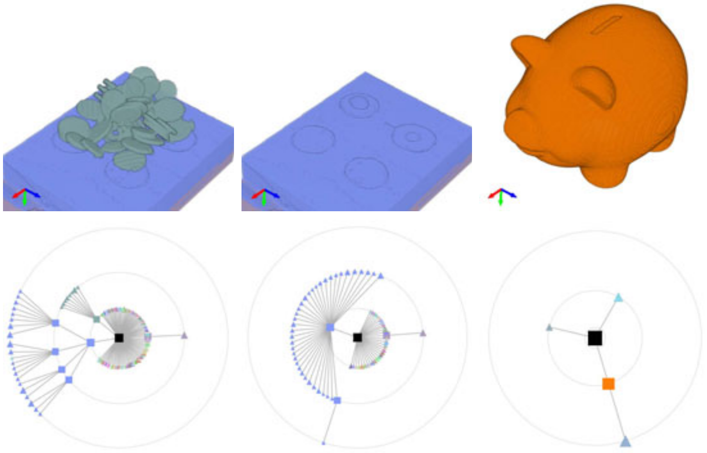
The Foot contains bones (bright objects) inside soft tissue (lower gray values), and it is surrounded by background noise. Removal of noise and also soft tissue can be easily achieved by both the max-tree and the watershed tree. Figure 1 shows results obtained with the Elongation criterion, and the resulting simplified trees. Since the max-tree encodes flat zones at each gray level, many branches are long, and they remain long also after filtering. Note that the layout is unevenly distributed due to a high branching factor near the root node, which leaves little space for the rest of the tree. The simplified watershed tree is more compact, which makes it easier to visually relate image objects to subtrees.

[1] sources: `volvis.org` and `www9.informatik.uni-erlangen.de/External/vollib`



(a) (b)

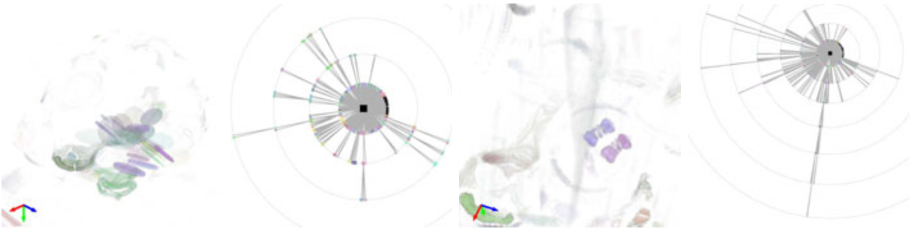
Fig. 1. Foot data set filtered with the Elongation criterion, and colored by the tree-based TF. (a) Max-tree, $\tau = 8.3$, direct rule. (b) Watershed tree, $\tau = 7.4$, $\tau_d = 6$.



(a) (b) (c)

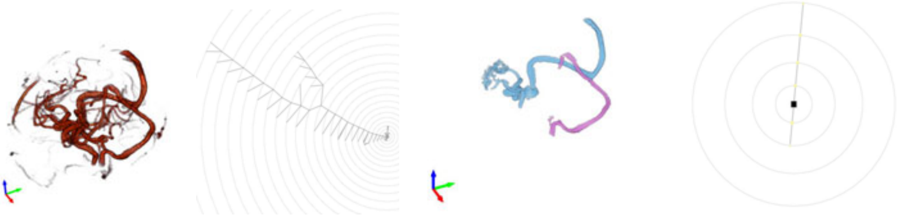
Fig. 2. Piggy bank data processed with the watershed tree. (a) Flat criterion, $\tau = 9.8$, $\tau_d = 5$. (b) Flat criterion, $\tau = 20$, $\tau_d = 5$. (c) Sparseness criterion, $\tau = 10.2$, $\tau_d = 4$.

The Pig consists of a hard material mounted on a wooden support. The inside contains a number of chocolate coins. There is almost no noise. The watershed tree does well at exposing the coins by filtering this data with the Flat criterion at a low threshold, see Fig. 2(a). At a higher threshold the coins disappear and only the support remains, see Fig. 2(b). Figure 2(c) shows that the Sparseness criterion can be used to remove all objects except the pig shape itself. The max-tree also manages to expose the coins using the Flatness criterion, and remove the objects surrounding them. This is shown in Fig. 3(a). Some coins have been removed, however, because they are connected to other objects, and do not fulfill the Flatness criterion. The max-tree holds most coins in separate branches, which is why the tree-based TF manages to color them differently. In the watershed tree, all coins are merged into one node. By using the Elongation criterion and max-tree filtering, two small springs in the Piggy bank's belly could be exposed (Fig. 3(b)), which was not possible with the watershed tree.



(a) (b)

Fig. 3. Piggy bank processed with the max-tree. (a) Flatness criterion, $\tau = 15$, subtractive rule. (b) Elongation criterion, $\tau = 5.5$, subtractive rule.



(a) (b)

Fig. 4. Angiogram processed by (a) Max-tree, Non-compactness criterion, $\tau = 2.5$, subtractive rule and (b) watershed tree, Sparseness criterion, $tau = 41.7$, $\tau_d = 9$

The Angiogram is very noisy, and the main purpose of filtering is to remove this noise, while retaining the shape of the blood vessels. The Non-compactness criterion in combination with max-tree filtering works very well for this purpose, as has been shown previously [22]. A result is shown in Fig. 4(a). The watershed tree can clearly not deal well with small structures, and only the large vessels can be retained, see Fig. 4(b).

The Tooth is quite challenging due to the high level of noise, which is very dense and present at high gray levels. The tooth itself consists of different materials, a very hard (bright) top, a hard (darker) body, and softer (even darker) material in the roots. Filtering the noise is achievable with the max-tree using the Flatness criterion and a subtractive filtering rule, see Fig. 5(a). Also, keeping the bright top can be done with the Non-compactness criterion, see Fig. 5(b). Separating the dark object inside the roots is not possible. The watershed tree is better at separating these structures from each other by some pruning and filtering with the Flatness criterion at various levels, see Fig. 6.

The min-tree can be used to obtain the dark object inside the roots, including the hard top and other parts, by considering the gradient magnitude image instead of the original image. This min-tree has 120,373 nodes, 96846 leaves, a depth of 255, and a branching factor of 1.07. A Sobel filter was used to compute smooth gradients. An example is shown in Fig. 7(a), in which the top part and soft root materials can be separated with the Flatness criterion. Figure 7(b)

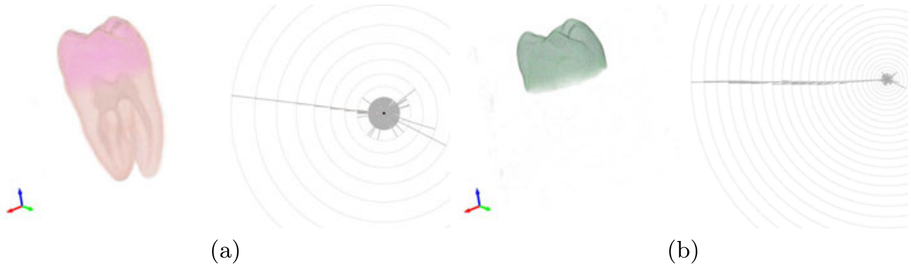


Fig. 5. Tooth processed by the max-tree. (a) Flatness criterion, $\tau = 1.8$, subtractive rule. (b) Non-compactness criterion, $\tau = 0.4$, subtractive rule.

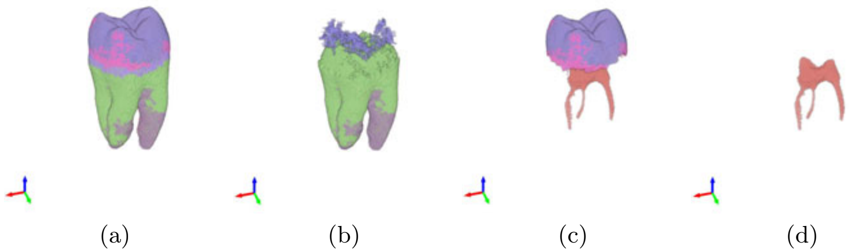


Fig. 6. Tooth processed by the watershed tree. The Flatness criterion was used to produce simplified results. (a) $\tau = 2.1$, $\tau_d = 8$. (b) $\tau = 2.1$, $\tau_d = 7$. (c) $\tau = 4.8$, $\tau_d = 8$. (d) $\tau = 6.1$, $\tau_d = 8$.

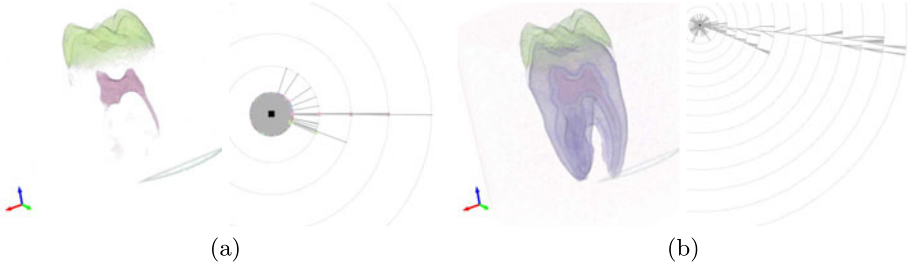


Fig. 7. Tooth gradient magnitude processed by the min-tree. (a) Flatness criterion, $\tau = 3.4$, subtractive rule. (b) Non-compactness criterion, $\tau = 0.4$, subtractive rule.

shows that the Non-compactness criterion can separate some more structures, though not as clearly as the watershed tree. We also experimented with min-trees constructed from the gradient-magnitude images of the other data sets, and achieved mostly worse results than with the original images. This means that features that could be extracted from the original data could not be extracted from the gradient-magnitude images.

6 Discussion

We have taken a data-driven approach for comparing the watershed tree and max-tree, two hierarchical representations for volumetric gray-scale images. The time complexity of constructing both representations is in principle the same, but the max-tree is faster to construct, because simpler and less computations are required.

Considering the *representation*, the max-tree has a fixed region model and merging order. Since the merging is strictly from high to low gray value in the max-tree, dark regions nested within light regions cannot be represented (this would require a min-tree). To some extent, this can be remedied by considering a gradient magnitude image, rather than the original image, because this type of nesting problem does not occur. However, the gradient is susceptible to noise, so some regularization is required. This may lead to loss of information, however, as we have seen in our experiments. The watershed tree is more flexible with respect to both region model and merging order, so the nesting problem does not occur at all. However, small regions cannot be represented well, due to the unreliability of computing region similarity for such small regions (cf. Eq. (1)). Since the merging process is greedy, errors made at the leaf level propagate through the tree, and undesirable merging may occur.

Simplification of the max-tree can only be done by attribute-based filtering, but it is flexible concerning the pruning strategies. In particular, the subtractive rule is attractive in that it allows for preservation of local contrast; there is no equivalent strategy for the watershed tree. An additional pruning method for the watershed tree is level selection, a form of region-based clustering, which cuts the tree at some depth and collapses the deeper nodes onto the surviving ancestors. In the max-tree, level selection becomes a trivial thresholding operation.

The watershed tree is a more *intuitive* representation than the max-tree. In the latter one, nodes correspond to peak components, and a node may represent multiple flat zones (‘regions’) that are not connected at that gray level. Also, the visualizations of the trees showed that max-trees are very unbalanced, which makes interaction with them difficult and correspondence between image region and subtree more difficult to understand.

Extension to more complex data types, such as vector-valued and tensor data, is possible with a watershed tree, because of the flexibility of region model and merging order. For the max-tree, however, it is not clear how to deal with non-scalar data, because a total ordering on the image elements would be required for constructing the tree. Some initial results have been obtained on color images [13,15].

In this paper, we have presented an initial comparison of two tree-based image representations. Our comparison has focused on practical aspects, such as flexibility, intuitiveness, and extendability of the max-tree (covering also min-tree and component-tree) and watershed tree. In future work, we plan to consider additional comparison criteria, and will include other tree structures known in the literature, such as the binary partition tree [17] and the level-line tree [12]. The binary partition tree uses criteria similar to those we use with the watershed tree to merge regions, yet the merging order differs, which results in larger and more complex trees. This tree may be able to better represent small regions than the watershed tree. A level-line tree (or inclusion tree) provides the basis for self-dual operators, and can be constructed by combining a min-tree and a max-tree. This tree can handle dark regions nested inside bright ones or vice versa simultaneously, in contrast to either the min-tree or the max-tree.

References

1. Armstrong, C.J., Price, B.L., Barrett, W.A.: Interactive segmentation of image volumes with Live Surface. Computers and Graphics 31(2), 212–229 (2007)
2. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall, New Jersey (1999)
3. Berger, C., Géraud, T., Levillain, R., Widynski, N.: Effective component tree computation with application to pattern recognition in astronomical imaging. In: Proc. IEEE Int. Conf. Image Processing 2007, San Antonio, Texas, USA, September 16-19, pp. IV–41–IV–44 (2007)
4. Beucher, S., Meyer, F.: The morphological approach to segmentation: the watershed transformation. In: Proc. ISMM 1993, pp. 433–481 (1993)
5. Breen, E., Jones, R.: Attribute openings, thinnings and granulometries. Comp. Vision and Image Und. 64(3), 377–389 (1996)
6. Cabral, B., Cam, N., Foran, J.: Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In: Workshop on Volume Visualization, Washington, DC, USA, pp. 91–98 (October 1994)
7. Levoy, M.: Display of surfaces from volume data. IEEE Computer Graphics and Applications 8(3), 29–37 (1988)
8. Marcotegui, B., Beucher, S.: Fast implementation of waterfall based on graphs. In: Proc. ISMM 2005, pp. 177–186. Springer, Heidelberg (2005)
9. Marcotegui, B., Zanoguera, F.: Image editing tools based on multi-scale segmentation. In: Proc. ISMM 2002, pp. 127–135. Springer, Heidelberg (2002)
10. Meyer, F.: Morphological multiscale and interactive segmentation. In: EURASIP Workshop on Nonlinear Signal and Image Processing, pp. 369–377 (1999)
11. Meyer, F., Maragos, P.: Multiscale morphological segmentations based on watershed, flooding, and eikonal PDE. In: Nielsen, M., Johansen, P., Fogh Olsen, O., Weickert, J. (eds.) Scale-Space 1999. LNCS, vol. 1682, pp. 351–362. Springer, Heidelberg (1999)
12. Monasse, P., Guichard, F.: Fast computation of a contrast invariant image representation. IEEE Trans. Image Processing 9(5), 860–872 (2000)
13. Naegel, B., Passat, N.: Component-trees and multi-value images: a comparative study. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 261–271. Springer, Heidelberg (2009)
14. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Trans. Image Processing 15(11), 3531–3539 (2006)
15. Passat, N., Naegel, B.: An extension of component-trees to partial orders. In: Proc. 16th Int. Conf. Image Processing (ICIP 2009), November 7-10, pp. 3981–3984. IEEE, Los Alamitos (2009)
16. Roerdink, J.B.T.M., Meijster, A.: The watershed transform: definitions, algorithms, and parallelization strategies. Fundamenta Informaticae 41, 187–228 (2000)

17. Salembier, P., Garrido, L.: Binary Partition Tree. as an efficient representation for image processing, segmentation, and information retrieval. IEEE Trans. Image Processing 9(4), 561–576 (2000)
18. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Trans. Image Processing 7(4), 555–570 (1998)
19. Salembier, P., Wilkinson, M.: Connected operators: A review of region-based morphological image processing techniques. IEEE Signal Processing Magazine 26(6), 136–157 (2009)
20. Urbach, E.R., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. IEEE Trans. Pattern Analysis and Machine Intelligence 29(2), 272–285 (2007)
21. Westenberg, M.A., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Volumetric attribute filtering and interactive visualization using the Max-tree representation. IEEE Trans. Image Processing 16(12), 2943–2952 (2007)
22. Wilkinson, M.H.F., Westenberg, M.A.: Shape preserving filament enhancement filtering. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, pp. 770–777. Springer, Heidelberg (2001)

Optimal Path: Theory and Models for Vessel Segmentation

Jean Stawiaski

Philips Healthcare, Medisys Research Laboratory, Suresnes, France

Abstract. This paper describes methods for vessel segmentation based on optimal paths. First, we recall a suitable algebraic framework for the optimal path problem on graphs through *Path Algebra*. We detail several popular models used for vessel segmentation and point out their limitations. Secondly, we present an extension of paths algebra which allows to solve constrained dynamic path problems. As examples, we detail an optimal path model with curvature constraints and one with dynamic time dependent costs.

Keywords: Optimal path, paths algebra, vessel segmentation.

1 Introduction

Optimal path is a popular technique used in many different areas of image processing and particularly for thread-like object segmentation. A nice theoretical framework for solving optimal path on graphs is given with path algebras [1]. Such an algebraic framework has already been used in the context of morphological segmentation by the watershed transform [2]. Path algebra unifies a large number of optimal path problems and furnishes the power of linear algebra methods for solving, understanding and designing optimal path models. In the first section of this paper, we use this framework to present several popular optimal path techniques used in the context of vessel segmentation. Unfortunately direct extensions of these techniques, including non-local path constraints, cannot be correctly handled in this framework. In such cases, appearing for instance in curvature dependent optimal path, a richer algebraic structure has to be introduced. This extension is the focus of the last part of the paper.

2 Optimal Paths on Graphs and Path Algebra

This section defines the notions of graph and some important related algebraic structures [1].

Definition 1 (Edge Weighted Graph). *An edge weighted graph G is a triplet $G = (V, E, W)$. The elements $v \in V$ are called vertices and the elements $e \in E \subset \{\{i, j\}, i, j \in V, i \neq j\}$ are called edges. W is a mapping of the edges E into $\mathbb{R}^+$. For each edge $e_{(i,j)}$ of G, $w_{(i,j)} = W(e_{(i,j)})$ is called the weight of $e_{(i,j)}$.*

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 417–428, 2011.

Definition 2 (Path). *A path $\pi_{(i,k)}$ is a sequence of edges: $\pi = (e_{(i,j)}, e_{(j,l)} ..., e_{(m,k)})$. $\pi_{(i,k)}$ is called a path from i to k. The set of all paths from i to k is denoted $\Pi_{(i,k)}$. We also write a path $\pi_{(i,k)}$ as a sequence of nodes $\pi = (i, j, ..., k)$, setting implicitly that for each pair of consecutive nodes, there exist an edge in G.*

Definition 3 (The Shortest Path Problem). *The shortest path problem between two nodes s and t consists in finding a path $\pi^*_{(s,t)}$ such that the sum of edges weights $\sum\limits_{e_{(i,j)} \in \pi^*_{(s,t)}} (w_{(i,j)})$ is minimal.*

The shortest path can be easily solved in the case of non-negative edge weights with Dijkstra's algorithm [3]. In the following, we move this path optimization problem in an algebraic framework in order to generalize and set limitations of what can be optimized on paths. Path algebra provides a very powerful theoretical framework because the shortest path becomes a solution of a linear equation in such spaces.

Definition 4 (Monoid (Semi-group)). *Let M be a set and $\oplus$ a binary operation on M. $(M, \oplus, \epsilon)$ is a monoid, if $\oplus$ is associative and ϵ is its identity element: $\forall a \in M, \epsilon \oplus a = a \oplus \epsilon = a$.*

Definition 5 (Dioid (Semi-ring)). *A dioid D, written $(D, \oplus, \otimes, \epsilon, \gamma)$, is an algebraic structure following the properties:*

1. *$(D, \oplus, \epsilon)$ is a commutative monoid. The identity element ϵ of $\oplus$ is called the zero of the dioid.*
2. *$(D, \otimes, \gamma)$ is a monoid. The element γ is called the unit element of the dioid.*
3. *The operation $\otimes$ is distributive with respect to the operation $\oplus$, and ϵ is an absorbing element for $\otimes$:*

$$\begin{array}{l} \forall (a,b,c) \in D^3, \;\; a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c) \;, \\ \forall (a,b,c) \in D^3, \;\; (b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a) \;, \\ \forall a \in D, \;\; a \otimes \epsilon = \epsilon \otimes a = \epsilon \;. \end{array} \tag{1}$$

Definition 6 (Path Algebra). *A path algebra $(P, \oplus, \otimes, \epsilon, \gamma)$, is an algebraic structure with the properties:*

1. *$(P, \oplus, \otimes, \epsilon, \gamma)$ is a dioid,*
2. *the operation $\oplus$ is idempotent: $\forall a \in P, \;\; a \oplus a = a$.*

Proposition 1 (The Natural Order). *The relation $a \prec b \equiv (a \oplus b = a)$ is an order in the monoid $(P, \oplus, \epsilon)$ if the law $\oplus$ is idempotent.*

The path algebra contains an order relation that allows to define the notion of minimizer over elements of P. In such spaces, the length of a path π is computed as $L(\pi) = \bigotimes\limits_{e_{(i,j)} \in \pi} w_{(i,j)}$. Dijkstra's algorithm [3] can be slightly modified to find optimal paths in such spaces by rewriting Bellman's optimality principle in the framework of path algebra:

$$L(\pi^*_{(s,j)}) = \bigoplus_{j \in \Gamma(i)} \left(L(\pi^*_{(s,i)}) \otimes w_{i,j}\right) , \tag{2}$$

where $\Gamma(i)$ is the set of neighbors of the node i. This formula points out that the optimal path $\pi^*_{(s,j)}$ can be found locally from the knowledge of $\pi^*_{(s,i)}$ and edge weights $w_{i,j}$.

Algorithm 1. Dijkstra's Algorithm

```
Require: Graph $G$, Vertex $s$, $\prec$, $\otimes$, $\epsilon$, $\gamma$
Ensure: Predecessor Map $PredMap$, Distance Map $Distmap$
  for all $v$ in $G$: $Distmap[v] := \epsilon$; $PredMap[v] := v$; $color[v] := WHITE$; end for
  $Distmap[s] := \gamma$ ; INSERT(Queue,$s$,$Distmap[s]$); $color[s] = GRAY$ ;
  while Queue is not empty do
    $u :=$ EXTRACT-MIN(Queue)
    for all vertex $v$ adjacent to $u$ do
      if $(w_{(u,v)} \otimes Distmap[u]) \prec Distmap[v]$ then
        $Distmap[v] := (w_{(u,v)} \otimes Distmap[u])$; $PredMap[v] := u$;
        if $color[v] = WHITE$ then
          $color[v] := GRAY$; INSERT(Queue, $v$, $Distmap[v]$);
        else
          if $color[v] = GRAY$ then
            DECREASE PRIORITY(Queue, $v$);
          end if
        end if
      end if
    end for
  end while
```

2.1 Models of Optimal Paths Used for Vessel Segmentation

We have now at our disposal a theoretical background for understanding and computing optimal paths. Numerous methods have already been proposed in the literature to encode vessel trees as a set of optimal paths. Popular existing methods for vessel segmentation are mainly based on an optimal path between two points. Several refinements also consider an extra dimension that encodes the radius of the vessels, anisotropic metric to ensure smooth path variations, subpixel accuracy using the fast marching method instead of graph based algorithms, and heuristics to encode the vessel smoothness and appearance. A review of these methods can be found in [4].

We consider now that a "vesselness" function $V_d : \mathbb{Z}^n \to [0, 1]$ is given [5]. In the following examples, we study several methods in order to extract a vessel tree or a particular vessel from given points lying on a vessel. We work on a graph $G = (V, E, W)$, where V is the set of pixels, $E \subseteq V \times V$ is the set of neighbors pixels. Assuming that V_d is high when the underlying point is likely to be a vessel, we consider the following edge weights:

$$\forall e_{(i,j)} \in E, w_{(i,j)} = (1 - \max(V_d(i), V_d(j))) \ . \tag{3}$$

Additionally to this simple scalar edge weight map, an anisotropic metric can also be taken into account if the main direction of the vessels is known:

$$\forall e_{(i,j)} \in E, w_{(i,j)} = \sqrt{\gamma_{i,j}^T T(i) \gamma_{i,j}} \ , \tag{4}$$

where $\gamma_{i,j} = \overrightarrow{(j-i)}$ is the vector representing the displacement from i to j. $w_{(i,j)}$ represents here the cost of traveling from i to j on a domain where the underlying metric tensor $T(i)$, aligned with the vessel axis, is supposed to be constant.

Optimal $\boldsymbol{L_1}$ Path: The first classical use of optimal paths consists in computing a shortest path according to the L_1 norm, where $L_1(\pi) = \sum_{e_{(i,j)} \in \pi} (w_{(i,j)})$. This formulation corresponds to optimal paths in the path algebra $(\mathbb{R}^+ \cup \{+\infty\}, \min, +)$.

A well known limitation of L_1 optimal path is the *shortcut* problem. When the searched path is relatively long, the path selection process is sensitive to small edge weight variations. The optimal path can thus locally cross edges of large weights corresponding to non-vessel regions.

Optimal $\boldsymbol{L_\infty}$ Path: In order to avoid the shortcut problem, a popular technique consists in computing an optimal path according to the L_∞ norm, where $L_\infty(\pi) = \max_{e_{(i,j)} \in \pi} (w_{(i,j)})$. This model corresponds to optimal paths in $(\mathbb{R}^+ \cup \{+\infty\}, \min, \max)$. In this case, the shortest path tree is also a minimum spanning tree.

The optimal L_∞ path is very sensitive to the quality of the weight map. In many practical situations with noisy or unperfect weight map, the L_∞ model fails to extract correctly the vessel tree. This limitation is called the *leakage* problem.

Optimal Path with "Oblivion": An interesting trade-off between the L_1 and L_∞ models has been proposed in [6]. In this model, the path length is computed as:

$$L_\alpha(\pi_{(i,j)}) = \sum_{t=0}^{kl} \alpha^t w_{(j-t,j-t-1)} \ , \alpha \in [0,1] \ , kl = L_0(\pi_{(i,j)}) \ . \tag{5}$$

where $L_0(\pi_{(i,j)})$ is the number of edges of the path $\pi_{(i,j)}$.

This definition presents several advantages, compared to the L_1 norm: long paths are less sensitive to shortcuts because the path length is highly dependent on the last edge weights of the path. Compared to the L_∞ norm, the proposed measure is less sensitive to leakage because edge weights are summed up on a large range. The algebraic structure representing this model is $(\mathbb{R}^+ \cup \{+\infty\}, \min, \oplus_\alpha)$, where $\oplus_\alpha$ is defined by:

$$L_\alpha(\pi_{(i,j)}) \oplus_\alpha w_{(j,m)} = w_{(j,m)} + \sum_{t=1}^{kl+1} \alpha^t w_{(j-t-1,j-t-2)} \ . \tag{6}$$

Unfortunately $(R^+ \cup \{+\infty\}, \min, \oplus_\alpha)$ is not a path algebra, since the order relation $\leq$ is not compatible with $\oplus_\alpha$:

$$\Big(L_\alpha(\pi_{(i,j)}) \leq L_\alpha(\pi'_{(i,j)})\Big) \nRightarrow \Big(L_\alpha(\pi_{(i,j)}) \oplus_\alpha w_{(j,m)} \leq L_\alpha(\pi'_{(i,j)}) \oplus_\alpha w_{(j,m)}\Big) \ . \tag{7}$$

Adding an edge to a path may decrease its length, which creates this incompatibility.

Since $(\mathbb{R}^+ \cup \{+\infty\}, \min, \oplus_\alpha)$ is not a path algebra, Dijkstra's algorithm can only give a rough approximation of the optimal path. Moreover the incompatibility of $\leq$ and $\oplus_\alpha$ makes the computation of optimal path intractable.

Optimal Path with Quasi Local Constraints, model 1: An interesting property of Dijkstra's algorithm is that the predecessor of a given node along the optimal path is kept in memory. This information can be used in order to add dynamic constraints depending on the last traversed edges. In the present model, edge weights of the graph are dynamically updated during the execution of Dijkstra's algorithm such that the length of a path is computed as:

$$L_\kappa(\pi_{(i,j)}) \oplus_\kappa w_{(j,m)} = L_1(\pi_{(i,j)}) + w_{(j,m)} + \kappa_k(m, j, ..., j-k) \ , \tag{8}$$

where $\kappa_k(m, j, ..., j-k)$ is a penalty measured from the k predecessors of the node j along $\pi_{(i,j)}$. A possible constraint can for instance be computed from the tortuosity of the path:

$$\kappa_k(i, i-1, ..., i-k) = 1 - \frac{d(i, i-k)}{k} \ , \tag{9}$$

where $d(i,j)$ is the Euclidean length between i and j. Contrary to the previous model, L_κ and $\prec$ are here compatible. The idea to change dynamically the edge weights depending on the current growing paths is very useful in order to include prior knowledge, predictors and local constraints on the path. A large number of models can be built from this template.

Dijkstra's algorithm using dynamic cost updates can unfortunately not produce paths minimizing the L_κ model because the model is not local but depends on a certain number of edges along the path. In other words, Bellman's optimality principle cannot be used at the local scale of a single edge of the graph.

Partial Conclusion: The path algebra provides one framework for solving many different optimal path problems. Unfortunately, as soon as non-local measures are added in the path model, solutions provided by the classical Dijkstra's algorithm are sub-optimal. The next section aims at proposing an extension of the path algebra framework that solves this limitation.

2.2 A Framework for Constrained Optimal Path

As pointed out earlier, optimal paths based on non-local measures cannot always be computed with Dijkstra's algorithm (or with other algorithm such as fast marching). We consider now a kind of time parametrization along paths leading to *trajectories*. The computation of an optimal trajectory is not the same problem, since it includes a notion of motion along a path. We introduce thus a new algebraic structure that will handle this difference.

We associate to each edge $e_{(i,j)}$ a transition cost function $h_{(i,j)}$ giving the arrival time t_j to node j when starting at time t_i from a node i: $t_j = h_{(i,j)}(t_i)$. Starting from a given node s at time t_s, we search the minimum amount of time needed to reach another node i. Such a problem, where the weight of an edge is time dependent cannot be studied in the path algebra framework unless all functions $h_{(i,j)}$ have a simple analytic formulation. We introduce thus another algebraic structure that permits to solve this problem. Let us define $(S, H, \oplus, *)$ where:

- the set S is a node state set having an internal associative and commutative operation $\oplus$. In other words, S is associated to a commutative monoid, its neutral element is denoted ϵ. This set S contains information embedded in the nodes of the graph and plays the role of the phase space of a dynamical system.
- The set H is a transition set, a set of application of S into S. We also suppose that:
$$\begin{cases} h(a \oplus b) = h(a) \oplus h(b) \quad \forall h \in H, a \in S, b \in S \ . \\ h(\epsilon) = \epsilon \quad \forall h \in H \ . \end{cases} \tag{10}$$
H is the thus the set of endomorphisms of $(S, \oplus)$. The "identity" endomorphism is denoted e.
- The operation $\oplus$ on S induces an operation $\oplus$ on H defined by $(h \oplus g)(a) = h(a) \oplus g(a) \quad (\forall h \in H, g \in H, a \in S)$. Its neutral element is denoted h' which associates for all $a \in S$ the neutral element $\epsilon \in S$. We have thus: $(h \oplus h') = (h' \oplus h) = h \quad (\forall h \in H)$.
- We consider in H a second law denoted $*$ defined by $h \in H, g \in H, \ \ h * g = g \circ h$, where $\circ$ is the composition of applications. $*$ is an intern operation, is associative, and admits the unit endomorphism e as neutral element. Moreover, $*$ is distributive with the law $\oplus$. It admits h^* as an absorbing element:
$$\begin{cases} (g * h^*)(a) = h^*(g(a)) = \epsilon \quad \forall a \in S, g \in H \ . \\ (h^* * g)(a) = g(h^*(a)) = g(\epsilon) = \epsilon \quad \forall a \in S, g \in H \ . \end{cases} \tag{11}$$

$(H, \oplus, *)$ is a dioid of endomorphisms. From a computational point of view, this formulation is problematic because the node state set S may be infinitely large and in numerous applications endomorphisms and their compositions are only known through their action on the elements of S. Nodes of the graph will not only be associated to a single distance value, but will encode several arrival times from several different paths hypothesis. The node state set will be as large as the range taking into account to compute the path length.

A generalization of Dijkstra's algorithm can be developed in the case where there exist at least a total pre-ordering $\propto$ on S which is compatible with $\oplus$ and such that $h_{(i,j)}(t_i) \propto t_i$ [1]. The algorithm pseudo code is given below.

Algorithm 2. Generalized Dijkstra's Algorithm

```
Require: Graph $G$, Vertex $s$, $\prec$, $\otimes$, $\epsilon$, $\gamma$
Ensure: Node States $E$
  for all $v$ in $G$: $E_v := \epsilon$; end for
  $E_s = \gamma$ ; $X_1 = \emptyset$ ; $X_2 = \{s\}$ ;
  while $X_2 \neq \emptyset$ or $X_1 \neq V$ do
    Select $i \in X_2$ such that $(E_i \propto E_j)$, $\forall j \in X_2$
    if $i \notin X_1$ then
      $X_1 = X_1 \cup i$; $X_2 = X_2 \setminus i$;
    end if
    if $X_1 \neq V$ then
      for all vertex $j$ adjacent to $i$ do
        Compute $E'_j = E_j \oplus h_{(i,j)}(E_i)$;
        if $E'_j \neq E_j$ then
          $X_2 = X_2 \cup j$;
        end if
        $E_j = E'_j$;
      end for
    end if
  end while
```

A major difference of this generalized algorithm, compared to the classical one, is that a node enters the queue every time that its state changes, and not only if its distance decreases. An intuitive point of view of this property is that the set X_2 of queued nodes is not limited to a thin narrow band but can be arbitrarily large.

Optimal Path with Dynamic Time Costs: A first example of dynamic time dependent optimal path is described in [1] in the context of a financial investment program. The idea behind this model is to promote paths growing sufficiently fast and on the other side freezing paths growing slowly. In this model, the path length will thus be dependent on the number of edges traversed so far.

If T is the maximum possible number of edges along a path, we define the nodes state set S as the set of vectors of size $(T+1)$ in $\mathbb{R}^+ \cup \{+\infty\}$. If $a = (a_0, ..., a_t)$ and $b = (b_0, ..., b_t)$ are two node states, we define $d = a \oplus b = (min(a_0, b_0), ..., min(a_i, b_i), ..., min(a_t, b_t))$. We define the transition function h by:

$$\begin{cases} h_{(i,j)}(a) = b \ . \\ with \ \ b_0 = +\infty \\ and \ \ b_t = a_{t-1} + \alpha^{t-1} w_{(i,j)} \quad for \quad t = 1, ..., T \end{cases} \tag{12}$$

The path length is then computed as $\min\limits_{0 \leq t \leq T}(a_t)$ and the total pre-ordering $\propto$ on S is defined by:

$$(a_0, ..., a_T) \propto (b_0, ..., b_T) \Leftrightarrow \min_{0 \leq t \leq T}(a_t) \leq \min_{0 \leq t \leq T}(b_t) \; . \tag{13}$$

The node state set contains here different paths arrival times corresponding to paths of different length. The size of this state set may be very large since it is as large as the number of edges of the longest path in the graph.

This model can be seen as a formalization of the propagation freezing method described in [7], which permits to optimize the number of visited nodes during the search of an optimal path. This model is interesting in the context of vessel segmentation since the path propagation is optimized and limited to vessel areas.

Optimal Path with Quasi Local Constraints, model 2: We are now able to solve the example of constrained optimal path according to the L_κ model. Note that this problem was already studied in [8] and also used for image segmentation in [9,10]. Since local edge weights cannot encode path curvature, the idea of these previous works was to construct a "lifted" graph, where edge weights encode curvature of a set of neighbor edges of the original graph. Optimal paths with curvature constraints are then computed with the classical L_1 norm in this lifted graph. The drawback of the method is that the lifted graph size grows exponentially with the number of adjacent edges taken into account. In fact, this method was only used in the case of curvature constraints computed from pairs of edges. In the present framework, such a lifted graph does not have to be explicitly computed. Instead of the lifted graph, the different path hypothesis configurations are encoded in the node states.

Let $|C^l|$ be the number of configurations of possible adjacent edges of length l. We associate to each node i, a state set E_i of size $|C^l|$. E_i will encode the actual distance of the node i for each possible configurations of its predecessors. The state set E_i^k represents the distance of the node i with predecessors corresponding to the k^{th} possible configuration. Considering that the node j can be attained from a node i according to the k^{th} configuration of predecessors, the transition cost $h_{i,j}$ is defined by:

$$E_j^k \oplus h_{(i,j)}(E_i^k) = \min \left(\, E_j^k \; , \; E_i^k + w_{(i,j)} + \kappa_k(i, j, ..., j-k)) \, \right) \; . \tag{14}$$

where κ_k is the curvature corresponding to the k^{th} configurations of predecessors of j. The distance of the node j to the source s is obtained as $\min\limits_{0 \leq t \leq |C^l|}(E_j^k)$, leading to the total pre-ordering $\propto$ presented in the previous example. The special case of curvature constraints can thus correctly be handled in this framework.

The presented L_κ model is an interesting alternative to the construction of a "lifted" graph [9,10] and provides the same results. The explicit construction of a lifted graph is here not necessary, but one has still to encode for each node a state set which may also be memory consuming. Moreover this model can also be altered to find approximate optimal paths by limiting explicitly the size of the node state set, i.e. the number of possible configuration hypothesis taken into account to compute the optimal path as described in [8].

3 Results

Fig. 1 compares the distance map and the associated optimal paths obtained from several models presented previously. In fig. 2, we illustrate the differences

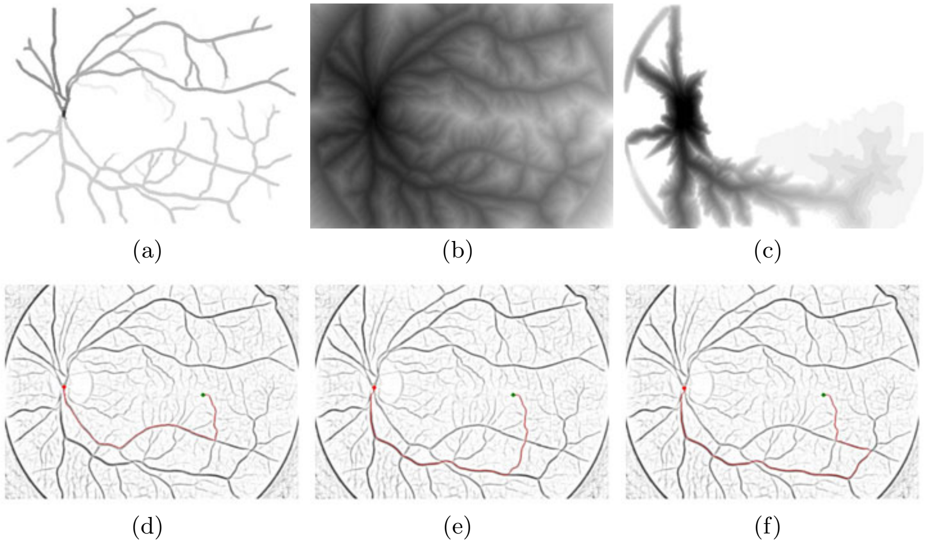


Fig. 1. (a) Distance map with L_∞ norm and (d) extracted path overlaid on V_d. (b) Distance map with curvature constraints (model 2) and (e) extracted path. (c) Distance map with time constraints and (f) extracted path.

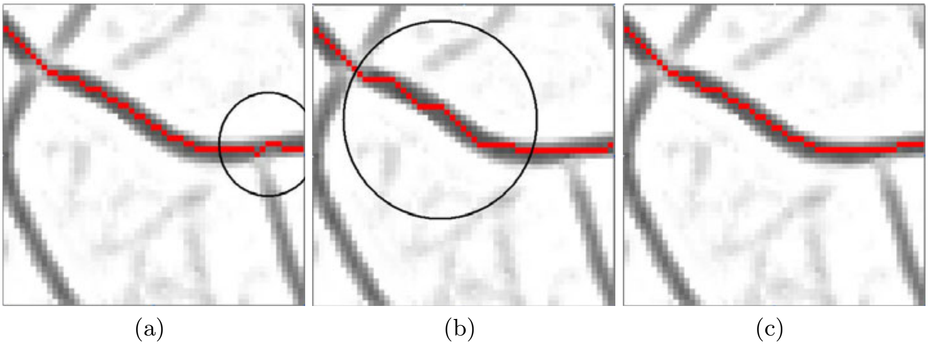
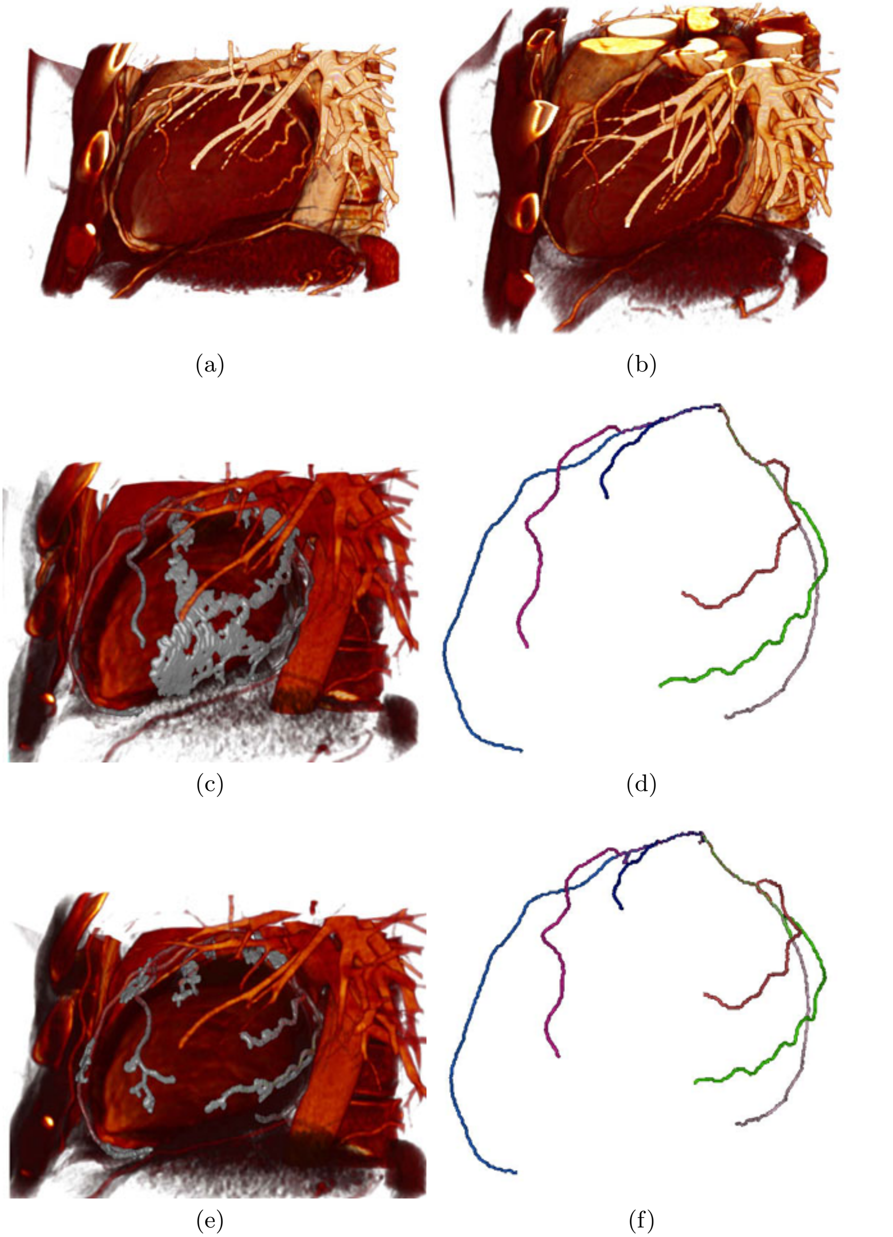


Fig. 2. (a) Shortest path with L_1 norm. (b) (Sub-)Optimal path with curvature constraints, model 1. (c) Optimal path with curvature constraints, model 2.



(a) (b)

(c) (d)

(e) (f)

Fig. 3. Left coronary artery tree segmentation. (a-b) Original computed tomography angiography. (c-d) Extracted vessel tree obtained from a minimal spanning tree (L_∞ model) rooted on the left coronary ostium. Gray surface indicates the space explored by the algorithm until all searched vessels are reached. (e-f) Extracted vessel tree obtained from the shortest paths with "oblivion". Gray surface indicates the space explored by the algorithm until all searched vessels are reached.

between the unconstrained and the curvature constrained models. Shortest L_1 path can locally present spurious trajectories whereas the constrained model 1 is very smooth but is composed of small straight steps. On the contrary, the curvature constrained model 2 is not sensitive to any of these limitations and furnishes the global optimizer of L_κ.

Fig. 3 illustrates an application of optimal paths to the extraction of the left coronary artery tree. In this example, we compare the L_∞ and the L_α (optimal path with "oblivion") optimal path models computed from user defined start and end points. This example highlights the practical efficiency of the L_α model. Whereas both models permit to extract correctly the searched vessel tree, the L_α model permits to minimize the number of voxels analyzed during the execution of Dijkstra's algorithm. This property is very important in practical applications since the size of the explored space is directly correlated to the computation time.

4 Conclusion

Path algebra is a powerful theory which provides one single framework for solving many different problems. The extension of this methodology to dynamic time dependent costs is also very useful in order to design new optimal path models for the segmentation of vessels. Future work will concentrate on applications of these models for complex vasculature vessel network segmentation.

The adaptation of this framework for general objects segmentation is also in our focus. Many morphological filtering and segmentation process, such as watershed segmentation [2], amoebas filtering [11], the image foresting transform [12], or geodesic image segmentation [13], heavily relies on the computation of a distance transform and associated optimal paths. The extension of classical optimal paths to non-trivial dynamic and time dependant models gives new means to design efficient solutions adapted to each particular application.

References

1. Gondran, M., Minoux, M.: Graphs dioids and semirings, new models and algorithms. Operations Research Computer Science Interfaces Series. Springer, Heidelberg (2008)
2. Meyer, F.: Grey-weighted, ultrametric and lexicographical distances. In: Computational Imaging and Vision, Mathematical Morphology: 40 Years On. Proceedings of the 7th International Symposium on Mathematical Morphology, vol. 30, pp. 289–298 (2005)
3. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
4. Lesage, D., Angelini, E.D., Funka-Lea, G., Bloch, I.: A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. Medical Image Analysis 13, 819–845 (2009)
5. Krause, M., Alles, R.M., Burgeth, B., Weickert, J.: Retinal vessel detection via second derivative of local radon transform. Technical Report No. 212, Department of Mathematics, Saarland University, Saarbrucken, Germany (2008)

6. Florent, R.: Image processing method and system, and medical examination apparatus for extracting a path following a threadlike structure in an image. European Patent (1999)
7. Deschamps, T., Cohen, L.D.: Fast extraction of tubular and tree 3d surfaces with front propagation methods. In: International Conference on Pattern Recognition, pp. 731–734 (2002)
8. Cowlagi, R.V., Tsiotras, P.: Shortest distance problems in graphs using history-dependent transition costs with application to kinodynamic path planning. In: Proceedings of the 2009 Conference on American Control Conference (2009)
9. Windheuser, T., Schoenemann, T., Cremers, D.: Beyond connecting the dots: A polynomial-time algorithm for segmentation and boundary estimation with imprecise user input. In: IEEE International Conference on Computer Vision, ICCV (2009)
10. Schoenemann, T., Cremers, D.: Globally optimal image segmentation with an elastic shape prior. In: IEEE International Conference on Computer Vision, ICCV (2007)
11. Lerallut, R., Boehm, M., Decencière, E., Meyer, F.: Image processing using morphological amoebas. In: Proceedings of the 5th International Symposium on Mathematical Morphology. Kluwers Academic Publishers, Dordrecht (2005)
12. Falcao, A.X., Stolfi, J., Lotufo, R.: The image foresting transform: Theory, algorithms, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 19–29 (2004)
13. Criminisi, A., Sharp, T., Blake, A.: GeoS: Geodesic image segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 99–112. Springer, Heidelberg (2008)

Cytology Imaging Segmentation Using the Locally Constrained Watershed Transform

N. Béliz-Osorio[1], J. Crespo[1], M. García-Rojo[2], A. Muñoz[3], and J. Azpiazu[1]

[1] Biomedical Informatics Group - LIA, DIA & DLSIIS
Facultad de Informática, Universidad Politécnica de Madrid
28660 Boadilla del Monte (Madrid), Spain
nbeliz@infomed.dia.fi.upm.es

[2] Department of Pathology, General Hospital of Ciudad Real
C/ Tomelloso s/n, 13005 Ciudad Real, Spain

[3] Department of Radiology, Facultad de Medicina
Ciudad Universitaria, 28040, Madrid, Spain

Abstract. The segmentation of medical images poses a great challenge in the area of image processing and analysis due mainly to noise, complex background, fuzzy and overlapping objects, and non-homogeneous gradients. This work uses the so-called locally constrained watershed transform introduced by Beare [1] to address these problems. The shape constraints introduced by this type of flexible watershed transformation permit to successfully segment and separate regions of interest. This type of watershed offers an alternative to other methods (such as distance function flooding) for particle extraction in medical imaging segmentation applications, where particle overlapping is quite common. Cytology images have been used for the experimental results.

Keywords: Image segmentation, image analysis, watershed, locally constrained watershed transform, medical images, cytology.

1 Introduction

In this paper, we are concerned with the segmentation of medical images. At present the semi-automatic or automatic analysis of medical images has become a fundamental tool in medicine[2]. For example: high resolution images of cancer cells can be used to determine the progression of cancer cells migration, indicating the invasion of cancer cells and cancer metastases [3], and the count of cells can help to diagnosis diseases [4]. As shown in [5,6,7,8], with the advancement of the new technologies is more common that hospitals have new technologies for testing and other assessments. This need for medical tools has generated a new line of research that include multiple disciplines (doctors, computer scientists, physicists, etc.) with the objective of effective analysis and management of biomedical data. The watershed transform is a well-known tool for segmentation of medical imaging [9,10,11]. The locally constrained watershed introduced by Beare [1] provides a great flexibility to control how shapes are extracted. Shape criteria can be added so that the gradient flooding that occurs

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 429–438, 2011.

in watershed methods is controlled. The technique possesses some characteristics found in other region based segmentation approaches, like geodesic active contours [12] and classical Partial Differential Equations (PDEs) [13,14]. As will be shown, the locally constrained watershed can successfully segment particles in some particularly difficult situations. This is especially relevant in medical image analysis, where precision is of high importance. We apply in this work the locally constrained watershed transform to biomedical images, particularly cytology images. These types of images often show fuzzy and overlapping objects, as well as non-homogeneous gradients. There are other methods that can also solve the problems caused by overlapping objects [15,16] and by touching edges are [17,18]. However, the aim of this work is to study the properties and capabilities of the locally constrained watershed transform in cytology imaging, especially in cases where the input image is directly segmented without prior preprocessing. The rest of this paper is structured as follows: Section 2 comments cytology imaging and its relevance. Section 3 treats the main points regarding the locally constrained watershed transform. Section 4 shows the application of the locally constrained watershed transform to cytology images and discusses some experimental results. Finally, Section 5 concludes the paper.

2 Cytology Imaging

2.1 Pleural Fluid Imaging

A cytology examination of pleural fluid (pleural fluid cytology) is a laboratory test that detects cancerous cells in the pleural space, the area that surrounds the lungs [19]. The test is performed by inserting a needle into the pleural space to draw off a sample of pleural fluid (the procedure is called thoracentesis). The sample of fluid is examined under the microscope for the presence of abnormal cells. About 75% of malignant pleural effusions are secondary to lung cancer, breast cancer and lymphoma. It is estimated that about 50% of patients with bronchial cancer and 40 to 50% of patients with disseminated breast cancer have a pleural effusion in the course of its evolution. Pleural metastases can also cause other primary tumors, such as ovarian carcinomas, sarcomas, melanomas, cervical cancer, cancers of the digestive tract, bladder, etc.

2.2 Papanicolau Smear Imaging

Papanicolau Smear (pap smear) is a medical test that uses five degrees to classify the cells of a cervix-vaginal cytology according to its degree of malignancy. In the test of Papanicolau, samples of cells of the vagina and the neck of the uterus are observed under the microscope. It allows to see cellular variations based on the hormonal cycle, if infections exist, and most important, the detection of malignant cells before tumors are pronounced. This test is able occasionally to detect the 95% of the cancers of the vaginal neck of the uterus, as well as others such as ovarian and endometrial cancers (although the reliability is smaller here). This test has radically changed the prognosis of women with cervical cancer.

3 Locally Constrained Watershed Transform

The main segmentation method in mathematical morphology [20,21,22] is the watershed transform [23]. The watershed transform is a segmentation technique that floods an input image gradient, which is considered as a topographic relief. The result of this process is a division of the input image in catchment basins, whose separation lines (watersheds) are the contours of the regions. The locally constrained watershed transform introduced by Beare [1] permits incorporation of constraints on the border of regions. Such region constraints[1] are imposed by using a cost function that depends on a shape (a structuring element). In the following, we will briefly focus on the foundations of how the region constraints are incorporated (please refer to [1] for a complete description and treatment of the method). A number of region segmentation techniques are able to include boundary constraints. In some cases those boundary constraints are essential for the satisfactory application of algorithms in real images. Energy minimization based methods of region segmentation, such as those using classical PDEs, are able to constrain border curvature. This is done by including a viscosity term in the energy function that modifies the rate of curve evolution. Careful selection of this term is often a critical factor in practical applications of these methods. There has also been some work on other types of boundary constraints in traditional region growing contexts. One method modifies the image gradient by applying a closing. It is based on a geophysical model in which a fluid is subjected to a variety of pressure [24]. A second method models the growing region as a polygon with the maximum edge length as the controlling parameter[25]. A technique called watersnakes demonstrates the energy minimization nature of the conventional watershed transform and includes border related terms explicitly by using a cost function [26]. Another approach within the watershed framework that can incorporate constraints is [27,28].

The locally constrained watershed transform introduced by Beare [1] provides a mechanism for higher level knowledge to be included in a conventional region growing framework. The particular higher level knowledge is the requirement that borders of regions should be, in some sense, smooth at all times during the growing process. This makes it possible to stabilize the growing process by preventing region leaks. This type of watershed considers how leaks may occur in the context of the watershed transform from the point of view of the physical model (gradient flooding) from which it was derived. This locally constrained watershed associates a cost to paths of translations of structuring elements, the so-called Minkowski paths.

Fig. 1 shows a step in a Minkowski path where the structuring element is a disk. Let B denote a structuring element, and let B_p symbolize B placed at pixel p (i.e., p is the origin of B). The cost of the Minkowski path of adjacent pixels $\pi = (p_0, p_1, \ldots, p_\ell)$ is then given by:

$$\hat{T}_f^\pi = \sum_{i=0}^{\ell-1} \widehat{\mathrm{cost}}(f, B_{p_i}, B_{p_{i+1}}), \tag{1}$$

[1] Each region (catchment basin) can have, if desired, each own particular constraint.

where $\widehat{\mathrm{cost}}(f, B_{p_i}, B_{p_i+1})$ measures the cost of translating the structuring element from pixel p_i to p_{i+1} in an image f. In the cost function of the locally constrained watershed, an auxiliary function, a summary function is used that is also a parameter of the method, along with the structuring element. Let us consider in the following the translation of a shape from pixel p to pixel q, which determines a translation vector $\mathbf{t} = \mathbf{q} - \mathbf{p}$. If we define

$$\mathrm{DS}_{\mathbf{p}}(B, \mathbf{t}) = B_{\mathbf{p}+\mathbf{t}} \backslash B_{\mathbf{p}}, \tag{2}$$

then the cost of moving the shape B from p to q is:

$$\widehat{\mathrm{cost}}'(p, q) = \max\{\Phi(\mathrm{DS}_{\mathbf{p}}(B, \mathbf{t})) - \Phi(\mathrm{DS}_{\mathbf{p}-\mathbf{t}}(B, \mathbf{t})), 0\}, \tag{3}$$

where Φ denotes a summary function that "summarizes" pixel values as a single value (a scalar). In the case where the size of the structuring element is a single pixel, the model is equivalent to the standard watershed transform. When the size is greater than one the region is constrained to be a union of the selected structuring element translated. The brief description above has commented one of the key ideas of the locally constrained watershed, the Minkowski cost, and also the two constraints that are associated to the image regions: (a) the shape (structuring element), and (b) the summary function. Such constraints will appear in the next section that discusses some experimental results. As summary functions, we will use the max and mean functions.

For instance, Fig. 2 provides a simple synthetic example that illustrates the capabilities of the locally constrained watershed to restrict the gradient flooding by using shape constraints. Fig. 2(a) displays an "idealized" situation where two structures touch. As will be shown in the next section, there exist numerous of such situations in cervical cytology images. First, we use the markers in Fig. 2(b) to extract both circles. Fig. 2(c) shows the result of the standard watershed transform. The locally constrained watershed can prevent the gradient flooding from the marked circle at the right to the one at the left (where there is no gradient wall) by introducing a shape constraint (particularly, a disk-shaped structuring element); both marked circles are successfully extracted in Fig. 2(d). The problem observed in Fig. 2(c) can be addressed, with the traditional watershed, by using, for example, the distance function [17]. The locally

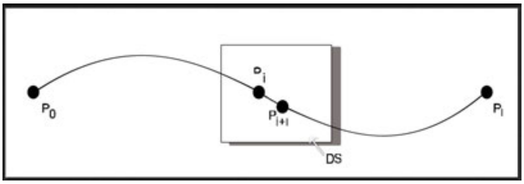


Fig. 1. A step in a Minkowski path

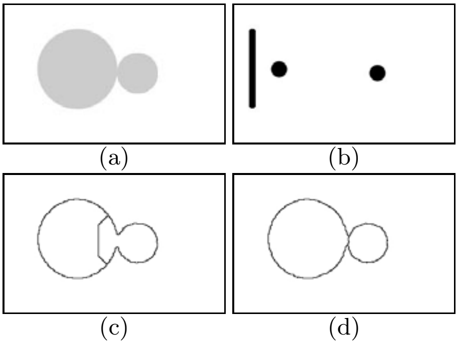


Fig. 2. A synthetic example of a segmentation of shapes with touching edges. (a) Original image. (b) Markers to extract for both circles. (c) Watershed transform result using (b) as markers. (d) Locally constrained watershed transform result using (b) as markers.

constrained watershed offers another alternative solution for particle extraction in medical imaging segmentation applications, where particle overlapping is quite common.

4 Experimental Results and Discussion

We have used images from "Hospital General de Ciudad Real" obtained using a scanner Aperio Scanscope XT (objective: 40×, resolution 0.25 microns per pixel, numerical aperture: 0.70). The images used are part of a set of 32 images. Of those, a subset of 24 presented the typical problems already mentioned (like overlapping, united cell membranes and incomplete gradient information) often found in medical images. Six cases have been selected and displayed in this paper. No preprocessing has been applied to the images prior to the computation of the morphologic gradient.

Fig. 3(a) shows a quite common situation in cytology: a cell that is out of focus, along with the other cells that are close enough to touch their cell membranes. Fig. 3(b) shows the successful outcome of the locally constrained watershed transform using a disk of radius 2 as a structural element, another disc of radius 1 for the background, and function $\Phi = max$. The locally constrained watershed transform adequately separates the object of interest of the image.

The cell that we want to segment in Fig.4(a) is blurred and is located besides another one, causing a weak gradient between the two cells. By using the locally constrained watershed transform we can prevent the leaks between the objects in the flooding process. The following configuration has been used in Fig.4(a): a disk-shaped structuring element with radius = 2 for both the object and for the background, and $\Phi = max$ as cost function.

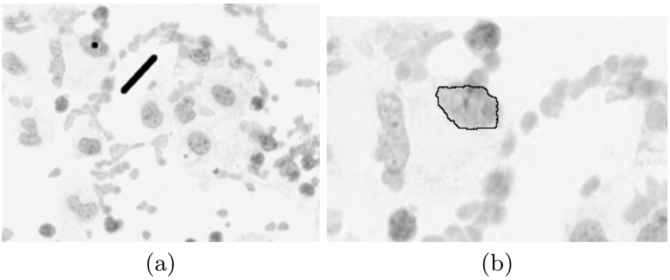


Fig. 3. (a) Original pleural fluid image with markers in the cell and in the background (size: 800×600). (b) Zoom of the locally constrained watershed transform result (cell regions constrained using a disk with radius = 2 and $\Phi = \max$). Note: the zoom image size in (b) is 300×200.

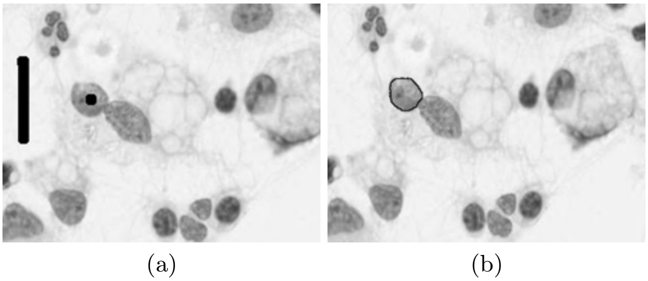


Fig. 4. (a) Original pleural fluid image with markers in the cell and in the background (size: 800×600). (b) Zoom of the locally constrained watershed transform result (cell regions constrained using a disk with radius = 2, background using a disk with radius = 2 and $\Phi = \max$).

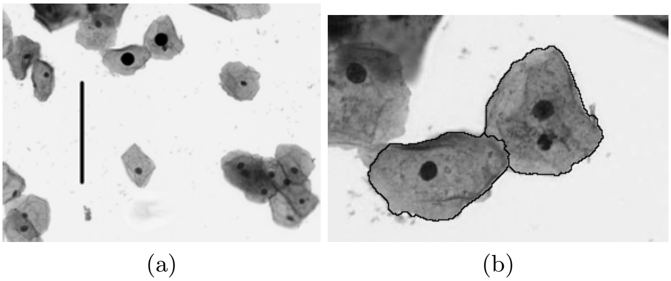


Fig. 5. (a) Original pap smear image with markers in the two cells. (b) Zoom of the locally constrained watershed transform result-cell regions constrained using a disk with radius = 2, background using a disk with radius = 1 and $\Phi = \max$.

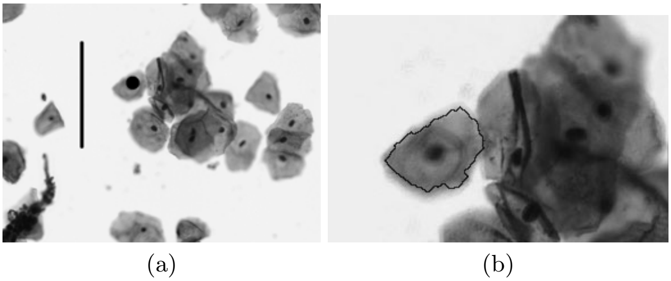


Fig. 6. (a) Original pap smear image with markers in the cell and in the background. (b) Zoom of the locally constrained watershed transform result (cell regions constrained using a disk with radius = 2, background using a disk with radius = 1 and $\Phi = \max$). Note: the zoom image size in (b) is 300×200.

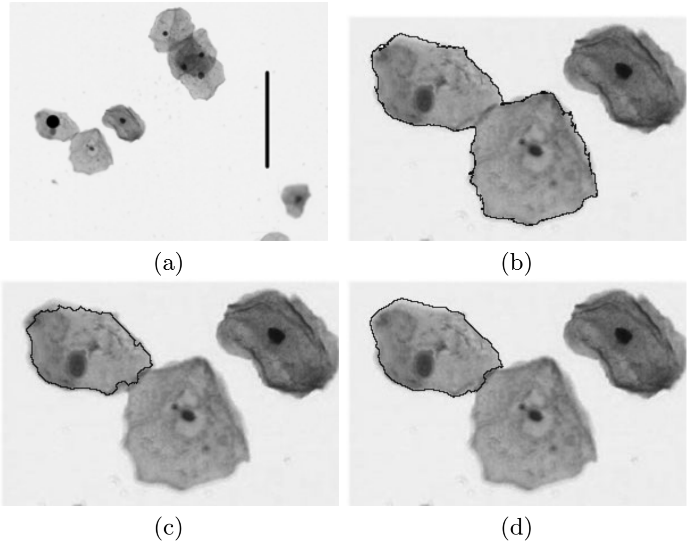


Fig. 7. (a) Original pap smear image with markers in the cell and in the background (size: 800×600). (b) Zoom of the watershed transform result. (c) Zoom of the locally constrained watershed transform result (cell regions constrained using a disk with radius = 2, using a disk with radius = 1, and $\Phi = \max$). (d) Zoom of the locally constrained watershed transform result (cell regions constrained using a disk with radius = 1, and $\Phi = \text{mean}$). Note: the zoom image size in (b) and(c) is 300×200.

In image Fig. 5(a) we can observe that the edges of the two cells that we want to segment are touching. The The locally constrained watershed obtains a suitable segmentation using a structuring element to constrain the gradient (Fig. 5(b)).

The example displayed in Fig. 6 is a similar case. Again, the shapes constraints of the locally constrained watershed can successfully extract the desired structure in Fig. 6(b).

In the case of Fig. 7(a), we can observe the effect that the locally constrained watershed summary function can have to refine a segmentation result. Fig. 7 (b) shows the result of the standard watershed transform. The locally constrained watershed successfully separates the marked cell (see Fig. 7(c) and 7(d)). In this example, two different summary functions have been used: $\Phi = \max$ (in Fig. 7(c)) and $\Phi = \text{mean}$ (in Fig. 7(d)). We can see that the use of the $\Phi = \max$ summary function produces somewhat irregular boundaries (Fig. 7(c)) and leaves some small portions of the target cell out (some parts of the background-cell gradient walls are quite weak). On the other hand, the summary function $\Phi = \text{mean}$ obtains a result with smoother edges (Fig. 7(d)).

5 Conclusion

In this paper we have applied the locally constrained watershed transform to segment cells in cervical cytology images. Results show that this type of watershed can successfully segment particles in difficult situations where there exist overlapping structures and the gradient is non-homogeneous. Thus, the locally constrained watershed transform offers a flexible and robust alternative that appropriately addresses this type of situations.

Acknowledgments

We want to thank S. Braendle[29] for his participation in the development of the software used to compute the paper results. This work has been supported in part by "Programa de Becas de Excelencia Profesional" of SENACYT-IFARHU of Panamá, by "Ministerio de Ciencia and Innovación" of Spain (Ref.: TIN2007-61768), and by ISCIII - "Subdirección General de Evaluación y Fomento de la Investigación" (PN I+D+I 2008-2011).

References

1. Beare, R.: A locally constrained watershed transform. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(7), 1063–1074 (2006), doi:10.1109/TPAMI.2006.132
2. Duncan, J.S., Ayache, N.: Medical image analysis: Progress over two decades and the challenges ahead. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 85–106 (2000), doi:10.1109/34.824822
3. Zhang, D., Xiong, H., Zhou, X., Yang, L., Wang, Y.L., Wong, S.: A confident scalespace shape representation framework for cell migration detection. Journal of Microscopy **231**(3) (2008) 395–407

4. Cseke, I.: A fast segmentation scheme for white blood cell images. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) Proceedings of the 11th IAPR International Conference on Pattern Recognition. Conference C: Image, Speech and Signal Analysis, vol. 3, pp. 530–533. IBBB, Berlin (1992), doi:10.1109/ICPR.1992.202041
5. Rojo, M.G., García, G.B., García, J.G., Vicente, M.C.: Preparaciones digitales en los servicios de anatomía patológica (i): Aspectos básicos de imagen digital. Revista Española De Patología 38(2), 69–77 (2005), http://www.pgmacline.es/revpatologia/volumen38/vol38-num2/38-2n02.htm
6. Rojo, M., Sánchez, F.: El impacto de la historia clínica electrónica en la investigación y la docencia. In (Coordinador), C.J. (ed.): De La Historia Clínica A La Historia De Salud Electrónica. Informes SEIS, vol. 5. SEIS, Pamplona: Sociedad Española de Informática de la Salud, pp. 315–345 Depósito legal: NA-183/2004 (December 2003)
7. Pablo, C., Lluís, J., Mata, X., Príncep, R., Naranjo, T.: Análisis cuantitativo de técnicas inmunohistoquímicas: Mejora de resultados mediante aplicación de software de análisis de imágenes digitales. In: Congreso Virtual Hispanoamericano de Anatomía Patológica, vol. 7 (October 2005)
8. Currie, W., Finnegan, D., Hamid, K.: 7. In: Integrating Electronic Health Record, 1st edn., pp. 135–182. Radcliffe Publishing (September 2009)
9. Grau, V., Mewes, A., Alcaniz, M., Kikinis, R., Warfield, S.: Improved watershed transform for medical image segmentation using prior information. IEEE Transactions on Medical Imaging 23(4), 447–458 (2004)
10. Klingler Jr., J., Vaughan, C., Fraker Jr., T., Andrews, L.: Segmentation of echocardiographic images using mathematical morphology. IEEE Transactions on Biomedical Engineering 35(11), 925–934 (1988), doi:10.1109/10.8672
11. Hamarneh, G., Li, X.: Watershed segmentation using prior shape and appearance knowledge. Image and Vision Computing 27(1-2), 59–68 (2009), doi:10.1016/j.imavis.2006.10.009
12. Murashov, D., Federation, R.: Method for segmentation of low contrast cytological images based on the active contour model. In: Shokin, Y.I., Potaturkin, O.I. (ed.) Automation, Control, and Information Technology. Signal and Image Processing, Novosibirsk, Russia, International Association of Science and Technology for Development, pp. 44–49 (June 2005), Hardcopy ISBN: 0-88986-461-6; CD ISBN: 0-88986-477-2.
13. Brockett, R.W., Maragos, P.: Evolution equations for continuous-scale morphological filtering. IEEE Transactions on Signal Processing 42(12), 3377–3386 (1994), http://ieeexplore.ieee.org/assets/img/btn.pdf-access-full-text.gif, doi:10.1109/78.340774
14. McOwen, R.C.: Partial Differential Equations. Tsinghua University Press, Beijing (2004)
15. Di Rubeto, C., Dempster, A., Khan, S., Jarra, B.: Segmentation of blood images using morphological operators. In: Proceedings of the 15th International Conference on Pattern Recognition, vol. 3, pp. 397–400. IEEE Computer Society, Washington, DC, USA (2000), doi:10.1109/ICPR.2000.903568
16. Srisang, W.: Segmentation of overlapping chromosome images using computational geometry. Computational Science, Walailak University, Nakhon Si Thammarat, Thailand. Krisanadej Jaroensutasinee, Contributor (December 2008)
17. Mohana Rao, K., Dempster, A.: Modification on distance transform to avoid over-segmentation and under-segmentation. In: Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8 International Symposium on VIPromCom, pp. 295–301 (2002)

18. Beucher, S.: Numerical residues. Image and Vision Computing 25(4), 405–415 (2007) (received September 23, 2005; revised June 26, 2006; accepted July 31, 2006), Available online September 26, 2006, doi:10.1016/j.imavis.2006.07.020
19. Davies, H.E., Sadler, R.S., Bielsa, S., Maskell, N.A., Rahman, N.M., Davies, R.J.O., Ferry, B.L., Lee, Y.C.: Clinical impact and reliability of pleural fluid mesothelin in undiagnosed pleural effusions. American Journal of Respiratory and Critical Care Medicine 180(5), 437–444 (2009), http://www.biomedsearch.com/nih/Clinical-impact-reliability-pleural-fluid/19299498.html, doi:10.1164/rccm.200811-1729OC
20. Serra, J.: Image Analysis and Mathematical Morphology, vol. I. Academic Press, London (1982)
21. Serra, J.: Image Analysis and Mathematical Morphology. Theoretical Advances, vol. II. Academic Press, London (1988)
22. Soille, P.: Morphological Image Analysis. Springer, Berlin (1999), http://web.ukonline.co.uk/soille
23. Beucher, S., Meyer, F.: 12. In: The Morphological Approach To Segmentation: The Watershed Transformation, pp. 433–481. Marcel Dekker, New York (1992), http://cmm.ensmp.fr/~beucher/publi/SB_watershed.pdf
24. Vachier, C., Meyer, F.: The viscous watershed transform. Journal of Mathematical Imaging and Vision 22(2-3), 251–267 (2005), doi:10.1007/s10851-005-4893-3
25. Beare, R.: Regularized seeded region growing. In: CSIRO Mathematical and Information Sciences, Locked Bag 17, North Ryde, Australia 1670, pp. 91–99. CSIRO Publishing (2002)
26. Nguyen, H.-T., Worring, M., Van Den Boomgaard, R.: Watersnakes: Energy-driven watershed segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(3), 330–342 (2003), doi:10.1109/TPAMI.2003.1182096
27. Vargas-Vázquez, D., Crespo, J.L., Maojo, V.: Morphological image reconstruction with criterion from labelled markers. In: Nyström, I., Sanniti di Baja, G., Svensson, S. (eds.) DGCI 2003. LNCS, vol. 2886, pp. 475–484. Springer, Heidelberg (2003), http://www.springerlink.com/content/n22peypd74j409xr/fulltext.pdf
28. Vargas-Vázquez, D., Crespo, J., Maojo, V., Ríos-Moreno, J.G., Trejo-Perea, M.: Reconstruction with criterion from labeled markers: new approach based on the morphological watershed. Journal of Electronic Imaging 19(4), 043001 (2010), doi:10.1117/1.3491494
29. Braendle, S.: Watershed algorithms with shape constraints. Master's thesis, Swiss Federal Institute of Technology - ETH Zurich, Switzerland (April 2008)

Stochastic Modeling of a Glass Fiber Reinforced Polymer

Hellen Altendorf[1] and Dominique Jeulin[2]

[1] Fraunhofer-Institut für Techno- und Wirtschaftsmathematik,
Fraunhofer-Platz 1, D-67663 Kaiserslautern, Germany
Hellen.Altendorf@itwm.fraunhofer.de
http://www.itwm.fhg.de/bv/

[2] Mines Paris Tech, Center of Mathematical Morphology,
35 rue Saint Honoré, F-77305 Fontainebleau Cedex, France
Dominique.Jeulin@cmm.ensmp.fr
http://cmm.ensmp.fr/

Abstract. Stochastic modeling of a material microstructure is in general composed of multiple steps. First, geometric properties of the sample are measured by image analysis. Second, an appropriate stochastic model is chosen and model parameters are estimated from the geometric properties. Third, additional characteristics are computed on the data set and on realizations of stochastic models to evaluate the quality of the fitting. In this article, we show how to measure geometric properties of a fiber system, estimate parameters for two different fiber models, and evaluate the realizations with orientation covariance and tortuosity of the fibers. The considered stochastic models are the newly developed bended fiber model, composed of a force-biased packing of ball chains, and the classical cherry-pit cylinder model. We show the advantages and limitations of both methods.

Keywords: Stochastic Modeling, Fiber Reinforced Composites, Fiber Separation, Parameter Estimation, Orientation Covariance.

1 Introduction

Fiber reinforced composites gain an increasing interest for industrial applications as the enclosure of aircrafts, boats and cars. Macroscopic properties of these materials are highly influenced by the geometry of the microstructure, in particular by the fiber directions. Physical properties can be optimized by adapting the parameters of the structural geometry. To this end, we need a realistic stochastic model including the main parameters for the fiber structure and taking into account the natural variability of the system. The realizations will be used as input in some simulation software of physical properties. We hope to optimize the microstructure with respect to their physical properties (for instance elastic moduli or thermal conductivity), changing the parameters of the stochastic model.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 439–450, 2011.

Fiber reinforced composites demand a stochastic model creating a non overlapping system composed of fibers with cylindrical cross sections and a low level of bending. The aim of this paper is to adapt the force-biased fiber packing [3] to a data set of a glass fiber reinforced composite and compare the realizations to those of a standard cylinder model.

The stochastic model for bended fibers represents fibers as ball chains. Random walks are used to initialize an overlapping system of bended fibers. A force-biased packing arranges the fibers such that they become non overlapping. This model can achieve high volume fractions up to approximately 72%.

Model parameters, as radius, volume fraction and orientation distribution, can be estimated directly from the original CT image [1]. Approximation of fiber length and curvature requests a separation of single fibers, which is applicable in the case of glass fiber reinforced composites with moderate fiber volume fraction and sufficient resolution [2].

For the evaluation of the models, we introduce a new method: the orientation covariance, which indicates the dependence of orientations in two image points with respect to their distance. We can interpret the covariance as a measure of local alignment. Moreover, the covariance yields the distance where fiber orientations become independent. Some studies of correlations of orientations between vectors are available in the literature ([9],[8], [7] and [12]). They are dedicated on statistics between a pair of unit vectors on a circle for the planar case, or on a sphere for the tridimensional case. None were proposed for the characterization of a random vector field, as is done in the present paper in quite a different and novel way. The statistical tools available in the literature are based on the standard correlation matrix between coordinates of the random vector.

We compare the bended fiber model to the classical approach of a cylinder model. Finally, advantages and limitations of both models as well as their fields of application are discussed.

2 Methods

Modeling fiber reinforced composites includes a set of methods: an appropriate stochastic model, estimators of the model parameters, and evaluation methods applicable to the real and the virtual material. First, we roughly explain the stochastic model presenting fibers as ball chains and being able to produce dense, hardcore systems of bended fibers. To estimate fiber parameters, we are making use of the single fiber separation presented in [2]. This segmentation results in a reconstruction of the fibers as ball chains and is therefore highly qualified for the parameter estimation, which is necessary for adapting the model. Estimators for those model parameters are presented.

To compare real and virtual material, we developed the orientation covariance, a modification of the classical covariance in image processing (see [18, p. 532]). Instead of evaluating the correlation of the binary image with respect to coordinate distance, we evaluate the correlation of the fiber orientations in foreground pixels. This approach can be applied to a segmented fiber system with the information of local orientation, as resulting e.g. from the algorithm introduced in [1]. Furthermore, we are comparing the tortuosity (see [6] and [16]) of the fibers, which will be measured on the separated fibers.

2.1 Fiber Model

The stochastic model considered in this paper is based on ball chains, initiated from a random walk and packed to a hardcore system with a force-biased approach. A fiber in the stochastic model is presented as a sequence of balls

$$P = \{p_1, \ldots, p_l\} \text{ with } p_i = (x_i, \mu_i, r_i) \in \mathbb{R}^3 \times S^2 \times \mathbb{R}^+, \tag{1}$$

consisting of the coordinate of the ball center $x_i \in \mathbb{R}^3$, an orientation $\mu_i \in S^2$ and a radius $r_i \in \mathbb{R}^+$. The orientation describes the local fiber orientation and the radius describes the local fiber radius. The main fiber orientation is chosen from a global orientation distribution defined for the system. We propose the β-distribution (see [17] or [15]) with a global parameter $\beta \in \mathbb{R}^+ \backslash \{0\}$. For $\beta = 1$ it results in the uniform distribution on the sphere, for $\beta \to 0$ the distribution concentrates on the z-axis and for $\beta \to \infty$ the orientations are distributed isotropically in the xy-plane.The probability density function of the β orientation distribution is

$$p(\theta, \phi|\beta) = \frac{\beta \sin\theta}{4\pi(1 + (\beta^2 - 1)\cos^2\theta)^{3/2}}, \tag{2}$$

where (θ, ϕ) are the polar coordinates of the orientation $\mu_0 \in S^2$.

A ball chain is created by a random walk starting from a random point in a cubic window with periodic boundary conditions. The orientation assigned to the first ball μ_1 is initiated with the main fiber orientation, chosen from the global orientation distribution. The orientation assigned to the i-th ball is distributed with the multivariate von Mises-Fisher distribution (see [10], [11] and [4]). The parameters of this distribution are two preferred directions and their reliability parameters κ_1 and κ_2. In our case, the preferred directions are the main fiber orientation μ_1 and the last chosen orientation μ_{i-1}. The level of bending is defined by the reliability parameters. The probability density function for the orientation distribution of the i-th ball is

$$f(\mu_i|\mu_1, \kappa_1, \mu_{i-1}, \kappa_2) = c(\mu_1, \kappa_1, \mu_{i-1}, \kappa_2)\ e^{\kappa_1 \mu_1^T \mu_i + \kappa_2 \mu_{i-1}^T \mu_i}. \tag{3}$$

The factor $c(x_1, \kappa_1, x_2, \kappa_2)$ serves for the normalization, such that the integral over S^2 is equal to 1. The normalization factor ist

$$c(x_1, \kappa_1, x_2, \kappa_2) = \frac{|\kappa_1 x_1 + \kappa_2 x_2|}{2\pi(e^{|\kappa_1 x_1 + \kappa_2 x_2|} - e^{-|\kappa_1 x_1 + \kappa_2 x_2|})}. \tag{4}$$

The radius r_i could be chosen from any distribution, in this paper we have chosen a fix radius for the system. The coordinates of the i-th ball are then defined by $x_i = x_{i-1} + \frac{r_i}{2}\mu_i$. This approach defines a overlapping system of bended fibers.

In a second step, we apply a force-biased approach, to achieve a hardcore configuration of the fiber system. Force-biased algorithms on spheres were introduced in [14] and statistically analyzed in [5]. The forces in our approach were inspired by the energy reducing models known from molecular dynamics [14] and describe the necessary displacement of the balls to relax the system. They do not act like mechanical forces. The algorithm works stepwise: In every step, forces are calculated according to the recent configuration and the balls are displaced with respect to their forces. Two kinds of forces are applied to the ball centers: repulsion and recover forces. The repulsion force arises in case of a fiber overlap and displaces the balls to an independent position. The recover force maintains the fiber structure. It keeps the distance and the angles between a ball and its neighbors, allowing only small deviation. The force conserving distances simulates springs between neighbor ball centers. The angle force simulates open springs between neighbor connections, which allows straightening of the fiber, but preserves fibers to bend in a clew. Both recover forces are provided with an initiating friction, which assures the stabilization of the packing process. The new configuration at the end of one step is defined by the displaced ball centers according to the sum of all forces.

2.2 Parameter Estimation

The separation of single fibers [2] results in a system of ball chains like the simulated one. We describe the configuration of the fiber system by

$$P = \{p_{1,1}, p_{1,2}, \dots, p_{1,l_1}, p_{2,1}, \dots, p_{n,l_n}\} \tag{5}$$

with $p_{j,i} = (x_{j,i}, \mu_{j,i}, r_{j,i}) \in \mathbb{R}^3 \times S^2 \times \mathbb{R}^+$. The fiber index is indicated with j and the balls in one fiber are ordered by the index i. Parameters for the stochastic model can be directly measured from this reconstructed microstructure. The estimation of the mean radius is evident. The mean fiber length is estimated from the sum of all fiber lengths divided by half the visible ends, excluding the cuts at image borders.

$$\hat{l}(P) = \frac{2}{\#(\text{visible ends})} \sum_{j=1}^{n} \sum_{i=2}^{l_j} |x_{j,i} - x_{j,i-1}| \tag{6}$$

The global fiber orientation distribution is determined by the parameter β, for which there are nowadays no estimators available. Still, it can be numerically approximated from the log-likelihood function:

$$ml(\beta) = n \ln \beta - n \ln(4\pi) - \frac{3}{2} \sum_{j=1}^{n} \ln(1 + (\beta^2 - 1)\cos^2 \theta_j), \tag{7}$$

for θ_j the angle between the j-th mean fiber orientation $\bar{\mu}(P, j) = \frac{p_{j,l_j} - p_{j,1}}{|p_{j,l_j} - p_{j,1}|} \in S^2$ and the global mean fiber orientation $\bar{\mu}(P)$, which can be derived by the main inertia axis of the set of all mean fiber orientations and their inverse: $M(P) = \{\bar{\mu}(P, j), -\bar{\mu}(P, j) \mid j = 1, \ldots, n\}$.

To estimate the curvature parameters κ_1 and κ_2, we study first the classical von Mises-Fisher distribution, which yields the relation ($\mathbb{E}$ being the mathematical expectation):

$$\mathbb{E}\left[\|\mu - \hat{\mu}\|^2\right] = \frac{2}{\kappa} + 2(1 - \frac{e^{\kappa} + e^{-\kappa}}{e^{\kappa} - e^{-\kappa}}). \tag{8}$$

with the second term disappearing for $\kappa \to \infty$. Already for $\kappa > 2$ the second term is negligible. The parameter κ can be numerically approximated from the empirical evaluation of

$$\mathbb{E}\left[\|\mu - \hat{\mu}\|^2\right] = \frac{1}{n} \sum_{i=1}^{n} 2(1 - \cos\theta_i). \tag{9}$$

The problem in estimating the parameters for the multivariate von Mises-Fisher distribution (mvMF) is the fixation of μ_1, while μ_{i-1} is varying during the process. Thus, for each realization of the von Mises-Fisher distribution parameters have changed. A fix mvMF distribution can be described as a classical von Mises Fisher distribution with a single reliability parameter $\kappa = |\kappa_1\mu_1 + \kappa_2\mu_2|$. For several realizations of mvMF with fix $\kappa_1, \mu_1, \kappa_2, \mu_2$ and known μ_1, μ_2 the estimator for κ would be trivial, and therefore κ_1 and κ_2 could be derived from the linear combination $\kappa\mu = \kappa_1\mu_1 + \kappa_2\mu_2$. Instead the random walk realizes $X_i \sim \text{mvMF}(\kappa_1, \mu_1, \kappa_2, \mu_{i-1})$. Thus, we achieve single realizations for varying distributions. Still, we can define variances to each preferred direction μ_1 and μ_{i-1} as:

$$\text{Var}_1(X) = \frac{1}{n-1} \sum_{i=2}^{n} \|\mu_i - \mu_1\|^2 \text{ and } \text{Var}_2(X) = \frac{1}{n-1} \sum_{i=2}^{n} \|\mu_i - \mu_{i-1}\|^2. \tag{10}$$

Simulations show that

$$\kappa_1 \approx 1/\text{Var}_1 \;\text{ and } \kappa_2 \approx 2/\text{Var}_2 \;\text{ for } \kappa_2/\kappa_1 > 2. \tag{11}$$

We are aware of the limitation of this estimator and that it can just be seen as a first guess of the parameters. Still, it is a measure of the curvature of the fiber. If we achieve similar measure values for the real and the virtual material, we can conclude that the model fitting is successful.

2.3 Measurements to Evaluate Realizations

The classical covariance ([19, pp.72], [18, pp. 532] and [13]) is defined on a stationary random set $B \subset \mathbb{R}^3$ as the probability that an arbitrary point x and its translation $x + hv$ are part of the set B. We can define the covariance with the following formula:

$$\text{Cov}_B(h, v) = \mathbb{E}[\mathbb{1}_B(\mathbf{0})\mathbb{1}_B(hv)], \text{ for } v \in S^2 \text{ and } h \in \mathbb{R}. \tag{12}$$

For a binary image $b : W \subset \mathbb{R}^3 \rightarrow \{0, 1\}$, we define B as the set of foreground points $B = \{x \in W \mid b(x) = 1\}$ and estimate the covariance as average over all image points $x \in W$: $\mathrm{Cov}_B(h, v) = \frac{1}{\|W\|} \sum_{x \in W} \mathbb{1}_B(x)\mathbb{1}_B(x + hv)$. The covariance is also known as normalized geometric covariogramm [13]. It evaluates the similarity of the indicator function on two points depending on their distance h in direction v. The covariance function starts at 1 for $h = 0$ and converges to the squared volume fraction for $h \rightarrow \infty$. From the evolution of the covariance function we can interpret the mean width of objects in a given direction and the existence and size of clusters. Comparison of covariance functions for different directions gives indications on the anisotropy of the microstructure.

The covariance can be applied to different characteristics by replacing the indicator function. In particular, the covariance of local fiber orientations can be defined on the scalar product of two orientations as follows:

$$\mathrm{Cov}_o(x, x + hv) = \mathbb{E}[(o(x)o(x + hv))^2 \mid \mathbb{1}_B(x)\mathbb{1}_B(x + hv) = 1], \tag{13}$$

where $o : W \rightarrow S^2$ is the local orientation map. By equation (13), we estimate a scalar, namely the conditional expectation of the square of the scalar product between two vectors located in x and in $x + hv$, knowing that the two points are in the set of interest. For a stationary random field of orientations, the covariance does not depend on the point x and can be written as

$$\mathrm{Cov}_o(h, v) = \mathbb{E}[(o(\mathbf{0})o(hv))^2 \mid \mathbb{1}_B(\mathbf{0})\mathbb{1}_B(hv) = 1], . \tag{14}$$

In the sequel of the paper, we assume stationarity. In that case, the covariance is estimated from the spatial average of the scalar product over the location of x in the image: $\mathrm{Cov}_o(h, v) = \frac{1}{\|W^*\|} \sum_{x \in W^*} (o(x)o(x + hv))^2$ with $W^* = \{x \in W \mid \mathbb{1}_B(x)\mathbb{1}_B(x + hv) = 1\}$.

It turns out, that this scalar equals the average of $\cos^2 \alpha$, α being the angle between the two vectors $o(x)$ and $o(x + hv)$. The squared cosine is one for equal orientations and drops down to zero for orthogonal orientations. The conditional expectation assures that only foreground points are taken into account. Otherwise, the curve would be a mixture of the classical covariance and the orientation covariance. The orientation covariance starts at 1 for $h = 0$ and converges to $1/3$ for $h \rightarrow \infty$ in an isotropically distributed fiber system (see appendix).

In contrary to the classical statistics on spherical data, the covariance also takes into account the spatial arrangement of the information. It can reveal local alignment and global anisotropy in the orientation distribution. It can be applied directly on data sets with the results from local orientation analysis as well as to realizations from the stochastic modeling. Therefore, it serves as a quality measure of the reconstructed material. Using the square of the scalar product enables a fast estimation by fast Fourier transform of the covariance of orientations for any separation hv. This is an advantage when dealing with images, which was not the scope of earlier statistical studies on populations of vectors.

In order to evaluate the bending of the fibers, we observe an additional characteristic: the tortuosity, which is defined as the ratio between geodesic fiber length and the distance of start and end point:

$$\hat{T}(P_j) = \frac{\sum_{i=2}^{l_j} |x_{j,i} - x_{j,i-1}|}{|x_{j,1} - x_{j,l_j}|}. \qquad (15)$$

For totally straight fibers (cylinders) the tortuosity is equal to 1. The higher the bending, the higher the tortuosity.

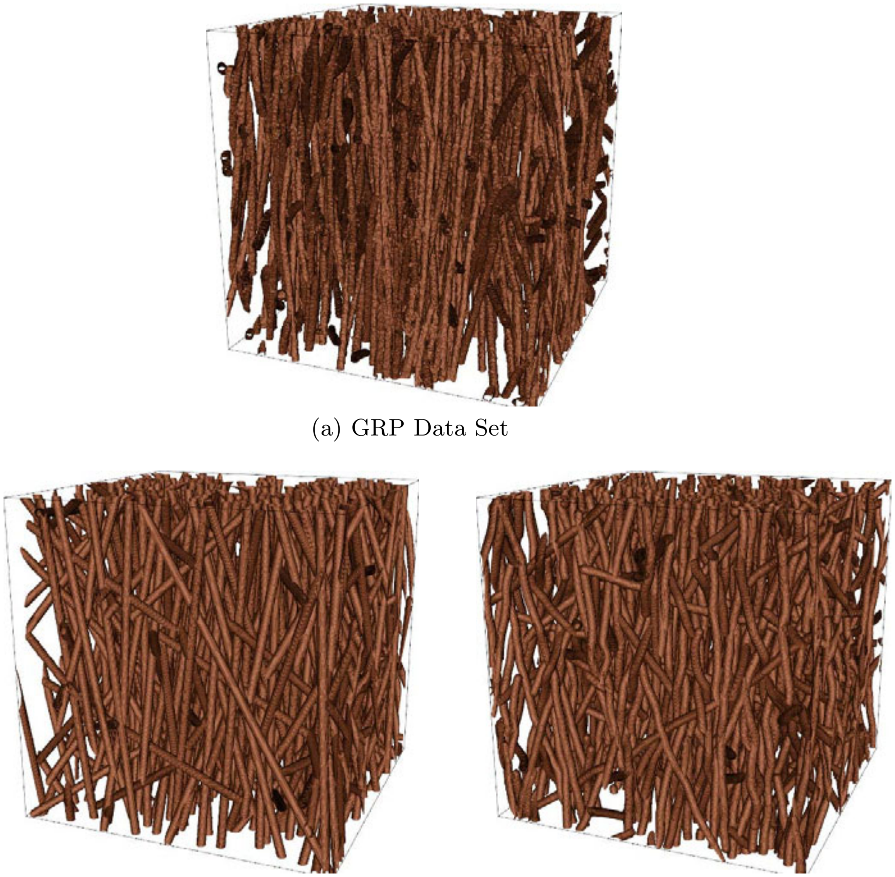
3 Application

For the adaptation of the stochastic fiber model, we have chosen a glass fiber reinforced polymer for which the single fiber separation was successfully applied. Fig. 1 shows a surface rendering of the binarized microstructure and realizations of the models. The fiber system shows only a low level of bending. The fitting of the bended fiber model is challenging in this case, as the deviation for the local bending tends to zero. Nevertheless, it is possible to trim the model to such a low bending and the model is unique in respecting all the measured geometric properties in a hardcore fiber systems.

We approximate the model parameters from the system of separated fibers according to the given estimators and create a realization of the bended fiber model respectively. Alternatively, we choose the cherry-pit cylinder model with RSA packing. With the given fiber length it is not possible to create a non overlapping cylinder system with the requested density. Overlapping conditions are lowered to a softshell-ratio of 0.7, meaning only the cores of the fibers with a radius of 30% of the real fiber radius are non overlapping. This reduction is a violation against the condition of hard glass fibers. Such a perturbation is not realistic for the given material. This is the first limitation of RSA cylinder models: long fiber systems can be realized only with a very low volume fraction and there exist far denser materials than the treated one, which requests 14.2% fiber volume fraction only. The second limitation concerns the edge treatment. The

Table 1. Estimation Results for real data sets and models

Parameters	Symbol	Image	Bended Fiber	Cylinders
Volume Fraction	V_V	0.142041	0.143878	0.141475
Mean Radius	$\bar{R}$	5.32035	5.32035	5.32035
Mean Length	$\hat{L}$	836.846	841.868	836.846
Mean Orientation	$\hat{\mu}$	(0.08, 0.09, 0.99)	(0.097, 0.087, 0.99)	(0.026, 0.067, 0.997)
Beta Estimate	$\hat{\beta}$	0.227	0.207	0.224
Kappa1 Estimate	$\hat{\kappa_1}$	56.8593	35.5017	∞
Kappa2 Estimate	$\hat{\kappa_2}$	356.694	395.628	∞
Tortuosity	$\hat{T}$	1.01188	1.0152	1



(a) GRP Data Set

(b) Cylinder Model

(c) Bended Fiber Model

Fig. 1. Modeling of glass fibre reinforced polymer. Surface Rendering of real data set and realizatons of stochastic models.

simulation of physical properties often requires periodic edge treatment. This condition cannot be realized for long cylinders in a small cutout, as cylinders tend to overlap themselves. The created realizations of the bended fiber and the cylinder model are shown in Fig. 1. The parameter estimation is given in Table 1 as well as the tortuosity as additional characteristic to evaluate the realizations. The bended fiber model fits well the true tortuosity. For straight cylinders, the tortuosity will naturally always equal one, which is too low in the present application.

The orientation covariance (Fig. 2) shows a local alignment of the fibers from the slow decrease in the first steps. In the present case, the asymptotic value of the experimental orientation covariance is close to 0.6. Furthermore, the elevation of the curve in z direction indicates the preference of this fiber direction. The asymptotic behavior of the experimental covariance reaching their sill for

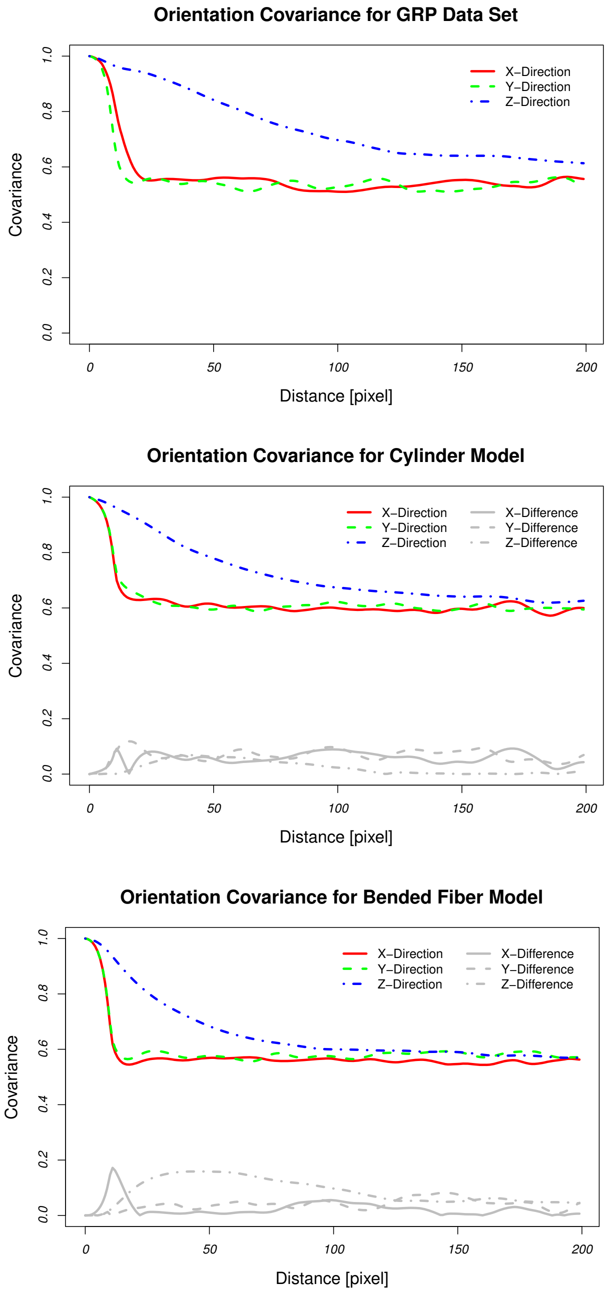


Fig. 2. Orientation Covariance for real data set and models. The model diagrams also include the difference in the orientation covariances to the real data set.

a finite range, is consistent with the stationarity assumption used in this study. We observe a similar shape for the curves in the cylinder model, but the level stays slightly too high for long distances. The local alignment of the bended fiber model fits the one of the real material, but local fiber directions are differing from the mean direction too fast. This causes the early decrease of the z direction curve compared to the real material. Furthermore, the relatively slow decrease of the z direction curve in the real material (compared to those in the virtual material) indicates probably a local clustering of directions. Such a cluster behavior has not yet been incorporated in the stochastic models. The bended fiber model probably includes local alignment of the fibers, but only for higher volume fractions. In a system with 14% volume fraction, fibers has too much freedom in their arrangement.

Furthermore, we assume that a single global orientation distribution is not suited well for this application. Instead of measuring a single concentration parameter β, one should try to adapt a mixture of β-distributions to the measured orientation distribution. Similarly every estimated mean parameter could be replaced by a distribution.

The low sensitivity of the covariance to a slight curvature of the fibers is reflected in the similarity of the two curves obtained for the simulation of cylinder and bended fiver models. The distinction between the two morphologies requires additional information, such as local curvature measurement. The parameters κ_1 and κ_2 in Table 1, as well as the tortuosity, reflect this distinction.

4 Conclusion

We presented the fitting of stochastic models to a fiber reinforced composite by parameter estimation and evaluated the created realization with the tortuosity and the orientation covariance. The advantages and limitations of both considered models are worked out. Bended fiber models need some more modification to create low-level bended fibers like present in the glass fiber reinforced composite. Cylinder models failed to give a non overlapping system of such long fibers, even if the requested volume fraction is quite low.

Perspectives are to extract information about fiber bending directly from the orientation covariance and the tortuosity. This would render the error-prone separation of single fibers obsolete. In particular for the case of carbon fiber reinforced composites, single fiber separation is very difficult.

Acknowledgement

This work is financially supported by the Institute Carnot M.I.N.E.S. and the Fraunhofer ITWM, Kaiserslautern.

References

1. Altendorf, H., Jeulin, D.: 3d directional mathematical morphology for analysis of fiber orientations. Image Analysis and Stereology (28), 143–153 (2009)
2. Altendorf, H., Jeulin, D.: Fiber separation from local orientation and probability maps. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009 Abstract Book, pp. 33–36. University of Groningen (2009)
3. Altendorf, H., Jeulin, D.: Random walk based stochastic modeling of 3d fiber systems (October 2010); accepted for Physical Review E
4. Banerjee, A., Dhillon, I.S., Ghosh, J., Sra, S.: Clustering on the unit hypersphere using von Mises-Fisher distributions. J. Mach. Learn. Res. 6, 1345–1382 (2005)
5. Bezrukov, A., Bargieł, M., Stoyan, D.: Statistical analysis of simulated random packings of spheres. Part. Part. Systems Char. 19, 111–118 (2002)
6. Decker, L., Jeulin, D., Tovena, I.: 3d morphological analysis of the connectivity of a porous media
7. Fisher, N.I.: Statistical analysis of circular data. Cambridge Univ. Pr., Cambridge (1996)
8. Fisher, N.I., Lewis, T., Embleton, B.J.J.: Statistical analysis of spherical data. Cambridge Univ. Pr., Cambridge (1993)
9. Jupp, P.E., Mardia, K.V.: A unified view of the theory of directional statistics, 1975-1988. International Statistical Review/Revue Internationale de Statistique 57(3), 261–294 (1989)
10. Karkkainen, S., Nyblom, J., Miettinen, A., Turpeinen, T., Pötschke, P.: A stochastic shape model for fibres with an application to carbon nanotubes (2008), http://newrobin.mat.unimi.it/OCS/index.php/ECS/ecs10/paper/view/62
11. Ko, D.: Robust estimation of the concentration parameter of the von Mises-Fisher distribution. Annals of Statistics 20, 917–928 (1992)
12. Mardia, K., Jupp, P., Ebooks Corporation: Directional statistics. Wiley, Chichester (2000)
13. Matheron, G.: Random sets and integral geometry. John Wiley & Sons, Chichester (1975)
14. Mościński, J., Bargieł, M.: The force-biased algorithm for the irregular close packing of equal hard spheres. Molecular Simulation 3(4), 201–212 (1989), http://www.informaworld.com/10.1080/08927028908031373
15. Ohser, J., Schladitz, K.: 3d Images of Materials Structures – Processing and Analysis. Wiley VCH, Weinheim (2009)
16. Peyrega, C., Jeulin, D., Delisée, C., Malvestio, J.: 3d morphological modelling of a random fibrous network. Image Analysis and Stereology 18, 129–141 (2009)
17. Schladitz, K., Peters, S., Reinel-Bitzer, D., Wiegmann, A., Ohser, J.: Design of acoustic trim based on geometric modeling and flow simulation for non-woven. Computational Materials Science 38(1), 56–66 (2006)
18. Serra, J.: Image analysis and mathematical morphology. Academic Press, London (1982)
19. Stoyan, D., Kendall, W.S., Mecke, J.: Stochastic Geometry and Its Applications. Wiley, Chichester (1987)

Appendix: Convergence of Orientation Covariance

The orientation covariance $\mathrm{Cov}_o(h,v) = \mathbb{E}[(o(\mathbf{0})o(hv))^2 \mid b(\mathbf{0})b(hv) = 1]$ converges for $h \to \infty$ to the covariance of two independent random vectors $x, y \in S^2$ distributed with the orientation distribution of the fiber system. In the case of isotropic orientation distribution, the orientation covariance converges to the same value for all $x \in S^2$. Thus, we can assume $x = (0,0,1)^T$ without loss of generality. Let θ and ϕ be the angles of the polar coordinates of y. The probability density function for the isotropic distribution is: $f(\theta,\phi) = \frac{1}{4\pi}\sin\theta$. The convergence of the orientation covariance for isotropic orientation distribution can be solved with partial integration as follows:

$$\mathrm{Cov}_o(h,v) \xrightarrow{h\to\infty} \mathbb{E}[(xy)^2] = \mathbb{E}[((0,0,1)^T y)^2] = \mathbb{E}[\cos^2\theta] \tag{16}$$

$$= \int_0^{2\pi}\int_0^{\pi} \cos^2\theta \frac{1}{4\pi}\sin\theta \; d\theta \; d\phi \tag{17}$$

$$= \frac{1}{2}\int_0^{\pi} \cos^2\theta \sin\theta \; d\theta \tag{18}$$

$$= \frac{1}{2}\left((-\cos^3\theta)|_0^\pi - \int_0^\pi 2\cos\theta\sin\theta\cos\theta \; d\theta \right) \tag{19}$$

$$= \frac{1}{2}\left(2 - 2\int_0^\pi \cos^2\theta\sin\theta \; d\theta \right) \tag{20}$$

From (18)=(20) follows $\int_0^\pi \cos^2\theta\sin\theta \; d\theta = \frac{2}{3}$ and therefore $\mathrm{Cov}_o(h,v) \xrightarrow{h\to\infty} \frac{1}{3}$.

Segmentation of Cracks in Shale Rock

Erik R. Urbach[1], Marina Pervukhina[2], and Leanne Bischof[1]

[1] CSIRO Mathematics Informatics and Statistics,
Locked Bag 17, North Ryde NSW 1670, Australia
Erik.Urbach@csiro.au, Leanne.Bischof@csiro.au
[2] CSIRO Earth Science and Resource Engineering,
PO Box 1130, Bentley, WA 6102, Australia
Marina.Pervukhina@csiro.au

Abstract. In this paper the use of morphological connected filters are studied for segmenting sheet- and thread-like cracks in images of shale rock. A volume formed from a stack of 2-D X-ray images is processed using 3-D attributes. The shape-preserving property of these filters provides accurate segmentation results while the use of rotation-invariant attributes allow robust and computationally efficient segmentation of cracks at all orientations. The results obtained using shape and size attributes are provided and discussed. The research presented here is part of a project with geologists to provide tools for automated segmentation and analysis of features of interest in various types of rock.

1 Introduction

Accurate crack segmentation is a key factor for petrophysical properties prediction from microtomograms. An aspect ratio (thickness-to-length ratio) of the cracks is important for choosing suitable parameters for effective medium models which are generally employed to obtain elastic properties of rocks, e.g., [2, 7, 8, 11]. Crack volume and surface roughness is crucial for analysis of stress dependency of elastic properties of rocks [12, 18] and even more important for understanding of their geomechanical response on stress like Mohr-Coulomb failure criterion, e.g., [16]. The exact 3-D shape of the crack is central to determine tortuosity which is a vital parameter in elastic conductivity, hydraulic permeability and poroelastic simulations [1, 3, 9]. The length and connectivity of the cracks and their number in a characteristic volume allows determination of fractal dimension of crack distribution and its percolation. e.g., [22]. The results of segmentation based on advanced image analysis methods presented in this study are a first and essential step to multi-scale quantitative analysis of petrophysical properties of rock from 3-D images.

Connected operators [5, 13, 14] are a commonly-used set of tools from the field of mathematical morphology [17, 20] for filtering [27], segmentation [19], and classification [23, 24] of features in 2-D and 3-D images. Besides being computationally efficient they are shape-preserving. In the case of binary images this means that they can either remove or preserve a connected component entirely

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 451–460, 2011.

but they cannot change its contours other than by merging it entirely with another connected component. This property also holds in the gray-scale case with respect to flat zones.

Operators that do not modify contours allow accurate segmentation of features of interest in images. In this paper these operators are used to automatically segment cracks in shale rock as part of a project with geologists to provide methods to characterize images of various kinds of rocks obtained with different imaging modalities. The performance of the method presented here is evaluated using a 3-D volume formed from 500 slices of 400x500 pixels.

The proposed method to automatically segment these cracks in shale rock uses attribute-based connected filters [13, 27]. To detect these cracks, which vary in shape between sheet-like and thread-like, we use a non-compactness[27] attribute and two attributes based on the lengths of the axes of a best-fit ellipsoid. Below the theory behind our method will be discussed first.

2 Theory

In the following discussion, image X is defined as a subset of the image domain $\mathbf{M} \subset \mathbb{R}^n$ (in this paper $n = 3$). The i^{th} connected component of X is denoted by C_i^X, where i runs over some index set I. A connected component in the level set $\mathcal{V}_h(f) = \{x \in \mathbf{M} | f(x) = h\}$ at level h of a gray-scale image f is known as a *flat zone*. At each gray level there may be multiple flat zones, which are denoted by $L_h^k(f)$, with k some index variable. Similarly, a *peak component* $P_h^k(f)$, where k runs over some index set I_h^f, is defined as the kth connected component of the threshold set $\mathcal{T}_h(f)$ of image f which is defined as

$$\mathcal{T}_h(f) = \{x \in \mathbf{M} | f(x) \geq h\}. \tag{1}$$

In this paper a grain will refer to a connected component in the binary and to a peak component in the gray-scale case.

A popular subset of connected operators are attribute filters [4] which preserve or remove grains based on whether the corresponding attribute values satisfy some criteria. The area opening [26] is an attribute filter with the criterion that the value of the area attribute must be at least λ pixels. For gray-scale images, attribute filters can be computed in a flexible and versatile way using tree-based representations [10, 13] of the flat zones in the input image. A computationally efficient algorithm for tree-based connected operators that is used in this paper is the Max-tree[13].

A Max-tree is a tree structure where node N_h^k represents peak component P_h^k in the image. For any two nodes N_{h1}^{k1} and N_{h2}^{k2} corresponding to the two peak components P_{h1}^{k1} and P_{h2}^{k2} the following holds:

$$P_{h2}^{k2} \subset P_{h1}^{k1} \iff h2 > h1 \wedge N_{h1}^{k1} \text{ is an ancestor of } N_{h2}^{k2} \tag{2}$$

and

$$P_{h1}^{k1} = P_{h2}^{k2} \iff k1 = k2 \ \wedge \ h1 = h2 \tag{3}$$

Filtering an image using the Max-tree involves three phases:

1. **Construction:** The Max-tree is constructed from the input image. Various efficient algorithms for this have been proposed [10, 13]. The algorithm used in this paper involves a recursive flooding procedure and a hierarchical FIFO queue as proposed by Salembier et al. [13].
2. **Filtering:** Nodes with attribute values that do not satisfy the criterion are marked for removal and are assigned a different gray value while all other nodes remain unchanged. This will be discussed in more detail below.
3. **Restitution:** In a single pass each pixel of the output image is assigned the intensity value of the corresponding node in the Max-tree.

The attributes and criteria used during the filtering phase can be increasing (such as area) or non-increasing (such as perimeter). In the latter case, a node that does not satisfy the criterion might have descendants that do satisfy the criterion. In those cases a strategy is needed to determine which nodes will be removed while maintaining a valid tree structure. Salembier et al. [13] described four strategies: Min, Direct, Max, and Viterbi while Urbach and Wilkinson [25] proposed the Subtractive rule. These strategies decide to remove a node N_h^k with attribute value $A(P_h^k)$ and criterion $A(P_h^k) \geq \lambda$ (and preserve otherwise) as follows:

Min: A node N_h^k is removed if $A(P_h^k) < \lambda$ or if one its ancestors is removed.
Max: : A node N_h^k is removed if $A(P_h^k) < \lambda$ and it has no remaining descendants.
Viterbi: The preservation and removal of nodes is considered as an optimization problem. For details the reader is referred to Salembier et al. [13].
Direct: A node N_h^k is removed if $A(P_h^k) < \lambda$, in which case its pixels are lowered in intensity to its highest ancestor that meets the criterion.
Subtractive: Similar to Direct rule, but this strategy also lowers by the same amount all the descendants of N_h^k.

3 Method

The cracks we want to segment in a 3-D volume of shale rock are recognizable as thin sheet- or thread-like structures with a lower intensity value than the average intensity of their rough-textured surroundings. Unfortunately, the intensity value of these cracks is not constant and similar intensity values are also abundantly present elsewhere in the volume, thus, a simple thresholding of the 3-D volume (or on the 2-D slices) will not yield good results. As the segmented cracks will be used for numerical analysis to characterize properties of the rock samples, each crack should be detected and segmented accurately. A variety of methods have been studied in the literature (such as [15]) for detecting thin structures in 2-D and 3-D. The method proposed here is based on connected filters as their shape-preserving property will give us very accurate segmentations. Furthermore, the use of rotation-invariant attributes avoids the need for computationally expensive probing at each and every orientation in 3-D.

Considering the 2-D slices of our 3-D shale rock volume, such as shown in Fig. 1 we can recognize cracks as thin "vessel"-like structures. As 2-D image processing is still more common than 3-D and visual inspection on screen so much easier, it seems obvious to try to detect cracks using 2-D algorithms and then combine the results afterward. Unfortunately, this yields poor segmentation results as pixels that belong to the same crack might appear fractured in 2-D and can even appear as singleton pixels. Therefore, the slices are considered as a single 3-D volume and the Max-tree with 3-D attributes is used as an attribute filter to segment the cracks.

The proposed method uses three attributes: non-compactness, length of the major axis, and the ratio of the lengths of the major and minor axes. The first is the 3-D equivalent of Hu's first moment invariant [6] and is normalized so that it has the minimum value of 1 for a perfect sphere and higher values as features become less compact. For the computation of this attribute and for the lengths of the axes we follow the computations as described by Westenberg et al. in [27]. They also describe the following two useful ratios:

$$\text{elongation:} \quad E(C) = |\frac{d_1(C)}{d_2(C)}| \tag{4}$$

and

$$\text{flatness:} \quad F(C) = |\frac{d_2(C)}{d_3(C)}|, \tag{5}$$

with d_1, d_2, and d_3 being the lengths of the major, mid, and minor axis respectively.

As the cracks appear to have sheet-like shape we considered using the flatness ratio as attribute. A closer inspection of the cracks present in the volume made clear that these cracks vary in size and shape between sheet-like and thread-like. If instead the ratio between d_1 and d_3 would be used as attribute, it would have higher values for features that have large d_1 and d_2 and a small d_3 (sheet-like) or that have large d_1 but small d_2 and d_3 (thread-like). This would also detect features with a mid-range value for d_2. Let us refer to this ratio as the crack attribute.

The third attribute we use is the length of the major axis d_1, which, unlike the non-compactness and crack attributes, is an increasing attribute. The crack attribute is the most significant of the three used to segment cracks. Non-compactness is very effective in detecting elongated structures even when they are curved and in removing all compact features. Defining a minimum length for the major axis excludes those that are too short to be considered a crack, regardless whether they are elongated or compact features.

4 Existing Methods

Existing methods were evaluated besides the proposed method. The easiest and most straight-forward approach to segment the cracks would be to use just simple thresholding such that the pixels with gray levels corresponding to cracks become foreground pixels while all other pixels become background pixels. Unfortunately, as can be seen in Fig. 1 the gray levels representing cracks are also dominant elsewhere in the volume.

A sophisticated method for finding linear features (such as cracks) in 2-D images was proposed by Sun and Vallotton [21]. It was originally developed to accurately trace neurite outgrowth. Although it works only for the 2-D case we could use it by segmenting the linear features (in our case cracks) in each of the 2-D slices separately and then combining the results into a 3-D volume afterward. As it can be assumed that a crack is present in a number of consecutive slices with similar size, shape, and position, we could turn this method into a "pseudo 3-D" approach where after the individual slices are segmented, only those segmented features are considered cracks that are present in multiple consecutive slices and with similar size, shape, and position. Given the output images Y_i created by Sun and Vallotton's method, we tried this "pseudo 3-D"-approach by computing the following steps for each output slice Y_i (for $0 \leq i \leq 499$):

1. Create a binary mask slice M_i by dilating the corresponding slice Y_i with a 15x15 rectangular S.E.
2. Compute $R_i = \min_{j=i-5}^{j=i+5} M_i$.
3. Compute $Z_i = Y_i$ AND R_i.

The resulting slices Z_i will only contain those features that exists in multiple consecutive slices in Y_i. The dilation with a rectangular S.E. is used to allow slight changes to the size and/or position of cracks between consecutive slices. This approach was very good at locating all the cracks as even the faintest cracks were detected. However, Sun and Vallotton's method cannot deal adequately with the large range of widths of the cracks that can be simultaneously present in a single slice. Furthermore, it introduced some curly artefacts as it tried to connect pieces of cracks obstructed by noise or some small feature. Our conclusion here is that robust and high quality crack segmentation results for Shale rock are only possible when true 3-D approaches are used as pieces of cracks often appear disconnected in a single 2-D slice but are connected in the 3-D case. Furthermore, note that cracks do not always have to be aligned perpendicular to the orientation of the slices and that a top or bottom of a crack might appear anything but elongated.

5 Experiments

The segmentation performance of our method was evaluated using an 8-bit grayscale 400x500x500 volume that was constructed from 500 2-D slices. The cracks were segmented by using the Max-tree with subtractive rule on the 6-connected volume. The results obtained by using the following parameters can be seen for slice 439 in Fig. 1: non-compactness ≥ 2.0, $d_1 \geq 40$, and $|d_1/d_3| \geq 14$. Preprocessing the volume with a Gaussian blur with $\sigma_x = 0.5$, $\sigma_y = 0.5$, and $\sigma_z = 1.0$ resulted in a bit smoother, visually more appealing segmentation result with a minor overall improvement in the segmentation accuracy. The latter can

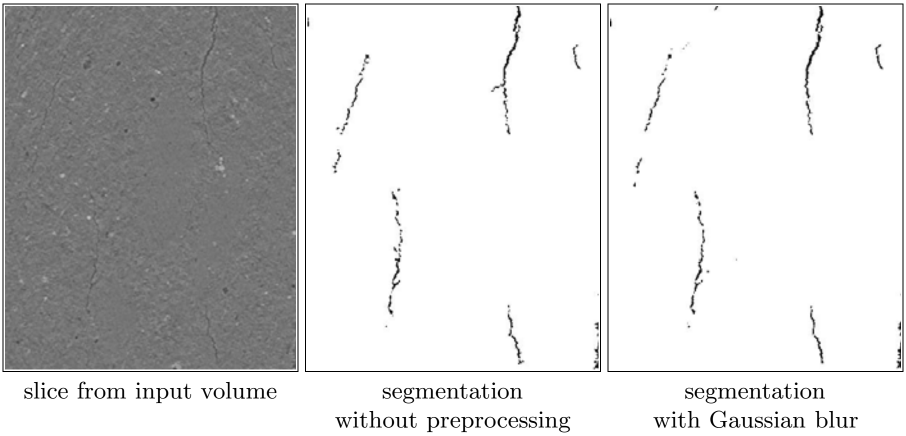

slice from input volume
segmentation without preprocessing
segmentation with Gaussian blur

Fig. 1. Effect of preprocessing input volume with a Gaussian blur before applying crack segmentation method

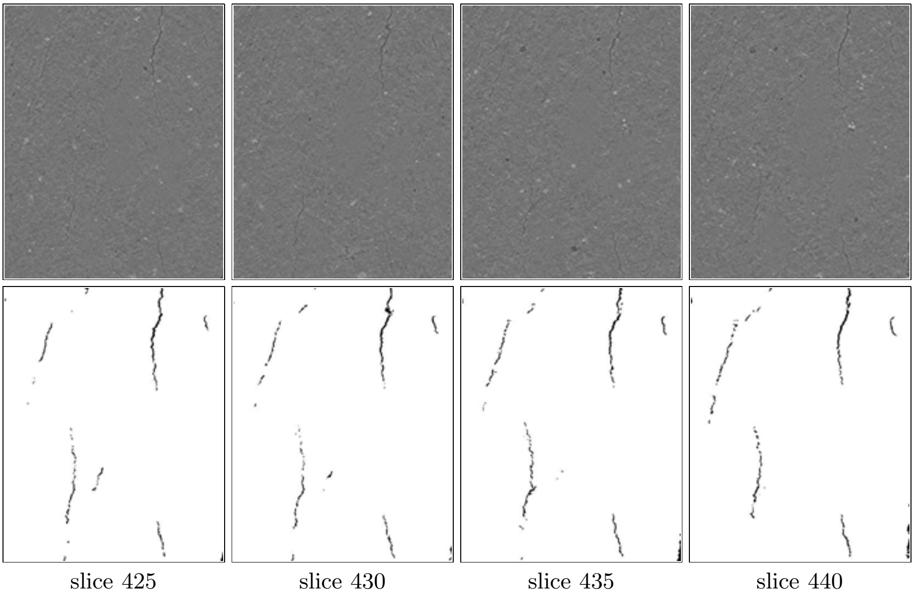


Fig. 2. Four 2-D slices from input volume (top row) and the corresponding slices from the output volume with the segmentation results (bottom row)

be explained by realizing that a very mild blur does suppress noise which affects the segmentation process as this noise can cause small gaps in cracks or cause non-related small features appear to be connected to a crack. Smoothing should however be limited to low amounts to ensure the characteristics of the cracks are not changed. Because of these advantages this mild Gaussian blur was used for preprocessing in the remainder of this paper.

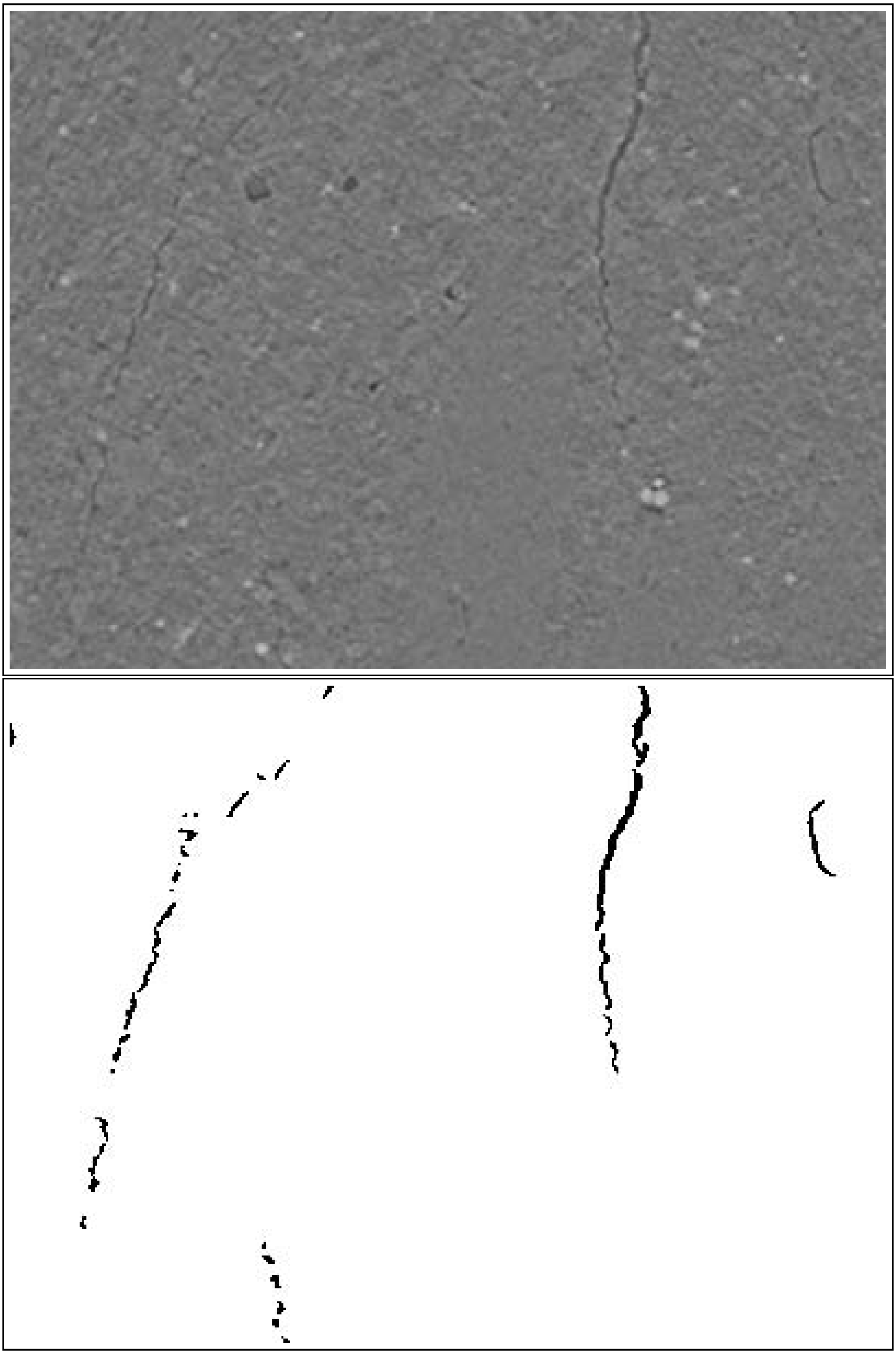


Fig. 3. Slice 440 from input volume (top) and the segmented cracks (bottom)

The filtered results were binarized by applying a threshold $\mathcal{T}_h(g)$ with $h = 4$ on the filtered output image g. This threshold value was chosen to remove noise and random features that are (almost) invisible and should thus be rejected.

Fig. 2 shows four slices of the input volume that are in close proximity of each other and the corresponding slices of the segmented result. Besides the accuracy of the segmentation results these slices also illustrate how features seen in a single slice as small as a dot can actually be part of a large crack and thus require the 3-D approach used here for crack segmentation. A section of the last of these four slices with the corresponding result image is shown enlarged in Fig. 3. Finally, the volume rendered segmentation results of the whole volume can be seen in Fig. 4.

Although the experimental validation would certainly benefit if we would have had access to more samples of Shale rock, the consistency of the segmentation results of our method throughout the whole volume gives some indication of the robustness and quality of the proposed method. The method was tested for several sets of parameter values with different values for: the attribute thresholds and the binarization thresholds and with and without Gaussian smoothing. All sets of parameter values with value close to the ones discussed before gave results similar to those presented in this paper. No improvement was noticed if 26-connectivity was used instead of 6-connectivity. By inspecting a few slices out of the volumes computed using each of these parameter sets, the best few were selected and the volume-rendered visualizations of these were then

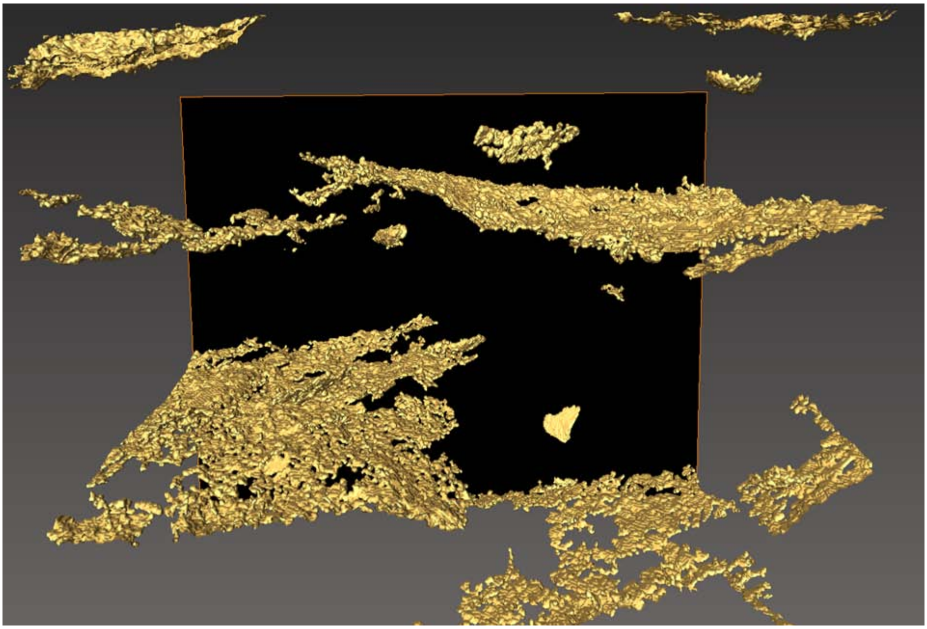


Fig. 4. Volume rendered segmentation result

carefully inspected manually by a domain expert assessing the quality, accuracy, and completeness of the segmented cracks.

6 Conclusions

A method for the application of crack segmentation in shale rock was presented. Morphological attribute filters with one size and two shape attributes were used. As we are unaware of any previously existing manually or automatically segmented results a quality assessment is limited at this point to visually inspecting the 3-D volume rendered result and individual slices of that same result. For a quantitative assessment of the quality of the segmentation results a ground truth would be needed. While a ground truth can be created relatively easily manually for 2-D images, doing this for even a single 400x500x500 volume would pose a rather challenging problem especially as voxel-level accuracy would be needed. As this method was developed due to a lack of a suitable existing method we are unaware of any existing method to do this automatically. We hope to obtain and use more 3-D rock samples to improve our assessment of the quality and robustness of the proposed method as well as the versatility of the proposed method for segmentation of other features in rock samples.

Evaluating the segmentation results slice by slice is limiting as many small and compact features that do not look like cracks clearly do belong to a crack when these areas are inspected in the 3-D volume rendered output. In the volume currently used, all cracks that we are aware of were detected by the proposed method while no false detections were spotted in the volume rendered result: both established by manual inspection by a domain expert. The segmented and binarized volume look very convincing when rendered in 3-D, unfortunately the same cannot be said when volume rendering is attempted on the input volume (either unprocessed or after inverting it) where the cracks have gray values that are also common in their heavily textured surroundings. This prevents an easy visual comparison between input and result.

References

1. Archie, G.E.: The electrical resistivity log as an aid in determining some reservoir characteristics. T. Am. I. Min. Met. Eng. 146, 54–61 (1942)
2. Berryman, J.: Long-wavelength propagation in composite elastic media. J. Acoust. Soc. Am. 69, 416–424 (1980)
3. Biot, M.A.: Mechanics of deformation and acoustic propagation in porous media. J. Appl. Phys. 33(4), 1482–1498 (1962)
4. Breen, E.J., Jones, R.: Attribute openings, thinnings and granulometries. Comp. Vis. Image Understand. 64, 377–389 (1996)
5. Heijmans, H.J.A.M.: Connected morphological operators for binary images. Comput. Vis. Image Understand. 73, 99–120 (1999)
6. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory IT-8, 179–187 (1962)

7. Kuster, G.T., Toksoz, M.N.: Velocity and attenuation of seismic-waves in 2-phase media.1. theoretical formulations. Geophysics 39(5), 587–606 (1974)
8. Kuster, G.T., Toksoz, M.N.: Velocity and attenuation of seismic-waves in 2-phase media.2. experimental results. Geophysics 39(5), 607–618 (1974)
9. Mavko, G., Nur, A.: The effect of a percolation threshold in the kozeny-carman relation. Geophysics 62(5), 1480–1482 (1997)
10. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Trans. Image Proc. 15(11), 3531–3539 (2006)
11. O'Connell, R.J., Budiansk, B.: Seismic velocities in dry and saturated cracked solids. Journal of Geophysical Research 79(35), 5412–5426 (1974)
12. Pervukhina, M., Gurevich, B., Dewhurst, D.N., Siggins, A.F.: Experimental verification of the physical nature of velocity-stress relationship for isotropic porous rocks. Geophysical Journal International 181, 1473–1479 (2010)
13. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Trans. Image Proc. 7, 555–570 (1998)
14. Salembier, P., Wilkinson, M.H.F.: Connected operators. IEEE Signal Processing Magazine 26(6), 136–157 (2009)
15. Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., Kinikis, R.: 3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. Medical Image Analysis 2, 143–168 (1998)
16. Schutjens, P.M.T.M., Hanssen, T.H., Hettema, M.H.H., Merour, J., de Bree, P., Coremans, J.W.A., Heilliesen, G.: Compaction-induced porosity/permeability reduction in sandstone reservoirs: Data and model for elasticity-dominated deformation. Spe. Reserv. Eval. Eng. 7(3), 202–216 (2004)
17. Serra, J.: Image Analysis and Mathematical Morphology, 2nd edn., vol. 1. Academic Press, New York (1982)
18. Shapiro, S.A.: Elastic piezosensitivity of porous and fractured rocks. Geophysics 68(2), 482–486 (2003)
19. Sofou, A., Tzafestas, C., Maragos, P.: Segmentation of soilsection images using connected operators. In: Int. Conf. Image Proc. 2001, pp. 1087–1090 (2001)
20. Soille, P.: Morphological Image Analysis: Principles and Applications, 2nd edn. Springer, New York (2002)
21. Sun, C., Vallotton, P.: Fast linear feature detection using multiple directional non-maximum suppression. Journal of Microscopy 234(2), 147–157 (2009)
22. Turcotte, D.L.: Chaos, fractals, nonlinear phenomena in earth-sciences. Rev. Geophys. 33, 341–343 (1995)
23. Urbach, E.R., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. IEEE Trans. Pattern Anal. Mach. Intell. 29(2), 272–285 (2007)
24. Urbach, E.R., Stepinski, T.F.: Automatic detection of sub-km craters in high resolution planetary images. Planetary and Space Science 57, 880–887 (2009)
25. Urbach, E.R., Wilkinson, M.H.F.: Shape-only granulometries and grey-scale shape filters. In: Proceedings of the ISMM 2002, pp. 305–314 (2002)
26. Vincent, L.: Grayscale area openings and closings, their efficient implementation and applications. In: Proc. EURASIP Workshop on Mathematical Morphology and its Application to Signal Processing, Barcelona, Spain, pp. 22–27 (1993)
27. Westenberg, M.A., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Volumetric attribute filtering and interactive visualization using the max-tree representation. IEEE Trans. Image Proc. 16(12), 2943–2952 (2007)

Size and Spatial Distributions Characterization of Graphite Nodules Based on Connectivity by Dilations

Luis A. Morales-Hernández[1], Ana M. Herrera-Navarro[1],
Federico Manriquez-Guerrero[2], Hayde Peregrina-Barreto[1],
and Iván R. Terol-Villalobos[2]

[1] Facultad en Ingeniería, Universidad Autónoma de Querétaro, 76000, México
luis_morah@yahoo.com

[2] CIDETEQ,S.C., Parque Tecnológico Querétaro S/N, SanFandila-Pedro Escobedo, 76700, Querétaro Mexico

Abstract. Microstructure in graphite nodules plays a fundamental role in mechanical properties in cast iron. Traditional measures used to study spheroid graphite are nodules density, nodularity, volume fraction and mean size. However, sometimes these parameters do not permit a good characterization of the microstructure since they do not allow the discrimination of different regions. In fact, other measures such as size and spatial distributions enable a better understanding of mechanical properties that can be obtained either by altering certain processing variables or through various heat treatments. In the present paper a method to characterize graphite nodules microstructure based on the connectivity generated by dilations is introduced. This approach, which takes into account size and spatial distributions of graphite, permits to relate the microstructure of graphite nodules with the wear behavior.

Keywords: Connectivity, Spatial and size distributions, Granulometry, Graphite nodules, Ductile cast iron, wear.

1 Introduction

It is well-known that the concept of connectivity plays an important role in image processing, particularly, its wide application in image segmentation and filtering. This concept inspired to Matheron and Serra to propose a new approach purely algebraic for Boolean lattices known as connectivity classes [12].

This new notion of connectivity overcomes some limitations of the traditional connectivity (formalized in topological or graph-theoretic frameworks), that restricts the type of objects to which it can be applied. Intensive work has been done on the study of this notion. Some recent studies dealing with this subject have been carried out in ([1,2,5,10,11,13,14,17,19], just to mention a few. In the present paper we are interested in the second generation of connectivity, particularly in that generated by dilations [1,2,10,12,13]. The main goal in the present work is to show the use of the connectivity generated by dilations to characterize

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 461–471, 2011.

the microstructure of ductile cast irons. Nowadays, ductile cast irons have been widely applied in the industry due to the low manufacturing cost, mechanical properties and easy fabrication.

The microstructure of ductile cast iron (frequently referred to as nodular or spheroid graphite iron) is characterized by the presence of spherical graphite particles or nodules throughout the material that nucleate on small inclusions during the solidification.

Most of published researches for ductile cast iron were devoted to study the microstructure, principally graphite nodules. Between the different measures used to characterize the microstructure, the nodule density, nodularity, volume fraction and mean nodule size, are traditional parameters in the characterization of the microstructure of ductile cast iron [6,8]. On the other hand, there are research works which have demonstrated that the relationship of the mechanical properties can be estimated by the size and shape of nodules [3,7,9,16,18]. Nevertheless, these works only study mean size, shape and density of nodules but not their spatial and size distributions.

The interest of the present paper is focused on the study of a ductile cast iron of an automobile part shown in Fig. 1. Particular attention is given to three regions of the mechanical part; two thick and one thin wall regions. Since the thin region cools faster than the thick regions it is expected to find some difference between the microstructure of regions that permits us to relate it with wear parameters.

This paper is organized as follows. One first gives a brief review of some morphological filters and the connectivity classes notion in Section 2. In Section 3 a study of the traditional parameters to characterize graphite nodules and the granulometry concept are used to correlate the microstructure with the wear behavior. Finally, in Section 4 one proposes the use of connectivity generated by dilations to understand better the relationships between spatial and size distributions and the wear behavior.

2 Some Basic Concepts of Morphological Filtering

2.1 Basic Notions of Morphological Filtering

Basic morphological filters in MM (mathematical morphology) [4,12,15] are the morphological opening $\gamma_{\mu B}$ and the morphological closing $\varphi_{\mu B}$ with a given structuring element μB, where B represents the elementary structuring element containing its origin; $\check{B}$ is its transposed set ($\check{B} = \{-x : x \in B\}$) and μ is a scale parameter. Thus, the morphological opening and closing are given, respectively, by equation (1):

$$\gamma_{\mu B}(f) = \delta_{\mu \check{B}}(\varepsilon_{\mu B}(f)) \quad \varphi_{\mu B}(f) = \varepsilon_{\mu \check{B}}(\delta_{\mu B}(f)) \tag{1}$$

where the morphological erosion $\varepsilon_{\mu B}$ and dilation $\delta_{\mu B}$ are expressed by $\varepsilon_{\mu B}(f)(x) = \bigwedge\{f(y) : y \in \mu\check{B}_x\}$ and $\delta_{\mu B}(f)(x) = \bigvee\{f(y) : y \in \mu\check{B}_x\}$. $\bigwedge$ is the inf operator and $\bigvee$ is the sup operator.

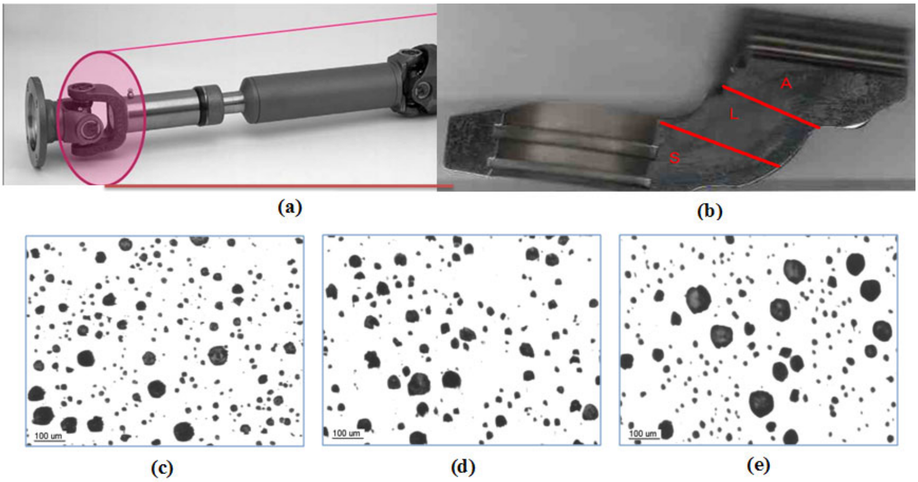


Fig. 1. (a) Automobile industry part, (b), (c) and (d) Images corresponding to regions A, L, and S

2.2 Connectivity Classes

One of the most interesting concepts proposed in MM is the notion of connectivity class. Let E be a non-empty space and $\mathcal{P}(E)$ the set of all subsets of E. In MM a connection, or connected class [12] on E is a set family $\mathcal{C} \subseteq \mathcal{P}(E)$ that satisfies the three following axioms

$$i/ \quad \emptyset \in \mathcal{C}, \qquad ii/ \quad x \in E \Rightarrow \{x\} \in \mathcal{C},$$

$$iii/ \quad \{X_i, i \in I\} \subseteq \mathcal{C} \quad and \quad \bigcap X_i \neq \emptyset \Rightarrow \bigcup X_i \in \mathcal{C}$$

An equivalent definition to the connected class is the point connected opening expressed by the following theorem:

Theorem 1. *(Point connected opening) The datum of a connected class $\mathcal{C}$ on $\mathcal{P}(E)$ is equivalent to the family $\{\gamma_x, x \in E\}$ [12] of the so called "point connected opening" such that*
iv/ for all $x \in E$, we have $\gamma_x(x) = \{x\}$,
v/ for all $A \subseteq E$, x, y in E, $\gamma_x(A)$ and $\gamma_y(A)$ are equal or disjoint,
vi/ for all $A \subseteq E$ and for all $x \in E$, we have $x \notin A \Rightarrow \gamma_x(A) = \emptyset$

An interesting connectivity can be built by means of dilatation [10,12,13]. Let $\mathcal{C}$ be a connectivity class in $\mathcal{P}(E)$ with connectivity opening γ_x. Assume that δ is an extensive dilation on $\mathcal{P}(E)$, then,

$$\mathcal{C}^{\delta} = \{X \in \mathcal{P}(E) : \delta(X) \in \mathcal{C}\} \tag{2}$$

is a connectivity class with $\mathcal{C} \subseteq \mathcal{C}^{\delta}$, and the corresponding connectivity openings are given by

$$\gamma_x^\delta(X) = \gamma_x(\delta(X)) \bigcap X \tag{3}$$

The example in Fig.2 illustrates this connectivity. At dilation 11 in Fig. 2(b) the output set is composed of two connected components of $\mathcal{C}^{\delta_{11}}$, whereas at dilation 32 in Fig. 2(c) the output set is composed of only one connected component of $\mathcal{C}^{\delta_{32}}$. Figures 2(d) and (e) illustrate the opening $\gamma_x^\delta(X) = \gamma_x(\delta(X)) \bigcap X$.

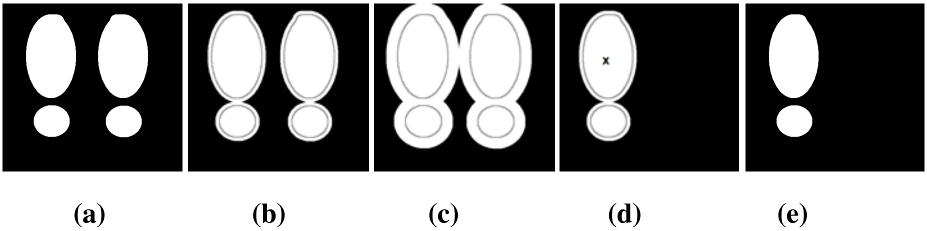


Fig. 2. (a) Original image X with 4 connected components of $\mathcal{C}$, (b), (c) Output images after dilation sizes 11 and 32,respectively, (d) Connected component $\gamma_x(\delta(X))$, (e) Connected component of $\mathcal{C}^{\delta_{11}}$

3 Graphite Nodules Characterization by Size Criterion

In the present section the wear behavior in a ductile cast iron of an automobile part is investigated. The study is focused on the use of traditional parameters and size distribution. It is known that the graphite, inside the metal matrix of the cast iron, behaves as a lubricant, then the wear resistance increases when the percentage of carbon also increases. However, this is not the only parameter involved in wear resistance as we will see below. To carry out the study, first, some wear tests were made on the mechanical part using a pin-on-disk tribometer CSM Instruments. Next, a set of representative images of the three regions, referred to as A, L, S, were acquired (20 samples were analyzed by each region). Two regions with the same thickness (A and S) and a thinner region (L) were selected as illustrated in Fig.1(a). Table 1 shows the wear results of the three regions to which the microstructure parameters of the samples will be related. Since region S had a minimum wear it was used as a reference to determine the wear percentage of the other regions.

3.1 Traditional Parameters

Traditional measures used to study spheroid graphite are nodularity, nodules density, volume fraction and mean size. Currently, the computation of the nodularity is based on how approximated the nodules shape to a circle are. For instance, a region of graphite is less nodular when the shape of the graphite nodules does not tend to be a circle. Even though there exist several measures to compute circularity, the most used one is the shape factor given by

$Sf = 4\Pi \cdot Area/Perimeter^2$. However, better results are obtained when using the conditional bisector as proposed in [8]. Once the circularity of nodules is determined, the nodularity is computed by $Nod = (CountCN/CountN) \times 100$, where $CountN$ and $CountCN$ are all the nodules and the spheroidal nodules, respectively. On the other hand, the density of nodules, which is defined as the number of nodules per unit area, is frequently used as an estimator of the nodule spatial distribution. However, this is a poor estimator since it is clear that spatial distribution is a more complex parameter as we will see below. Finally, the volume fraction is given by $Vf = NA/TA$, where NA is the nodule area and TA the total area (nodule area and matrix area). Table 1 shows the values of different parameters. Particularly, observe the percentages of nodularity for the three regions. On the one hand, it is observed that the nodularity of the three regions is good (greater than %80). On the other hand, by taking into account the nodularity values of the regions, the smallest value belongs to the region S and the greatest one to the region L. Contrarily, the region L has a greater wear than region S. Thus, even though nodularity is frequently used in characterizing graphite nodules microstructure, in the present study the shape of nodules does not play a main role. Between the other parameters, it is observed in Table 1 that the volume fraction of the three regions is very similar, showing that the amount of graphite is the same for the three regions. Thus, the percentage of graphite, inside the metal matrix, does not also play a main role in the wear resistance. Similarly, the mean size of nodules also has practically the same value for the three regions, then the effect in the wear behavior is neglected. The unique parameter which shows a correlation between wear and microstructure is the nodules density. Regions A and S with the same thickness and less wear, contain more nodules than the region L. Nevertheless, it is interesting to note that even if region L contains less nodules per unit area than regions A and S, the volume fraction and the mean size is practically the same. Then, the granulometry of region L must have a substantial difference with those of regions A and S.

Table 1.

Region	*Nod/Area*	*%Nodularity*	*% Volume fraction*	*Mean size* (μm)	*Wear*(%)
S	279	86.11	13.82	14.94	*Reference*
A	275	92.11	14.85	15.11	12
L	200	93.56	14.78	14.34	33.5

3.2 Size Distribution of Graphite Nodules

Let us study in this section the size distribution of nodules using the important concept of granulometry [12].

Definition 1. *A family of openings* $\{\gamma_{\mu_i}\}$, *where* $\mu_i \in \{1, 2, \ldots n\}$, *is a granulometry if for all* $\mu_i, \mu_j \in \{1, 2, \ldots n\}$ *and for all function* f, $\mu_i \leq \mu_j \quad \Rightarrow \gamma_{\mu_i}(f) \geq \gamma_{\mu_j}(f)$

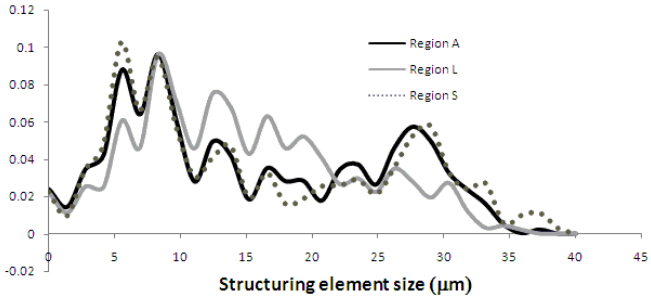


Fig. 3. Granulometries corresponding to regions A, L, and S

Since there exists overlapping between some graphite nodules, a granulometry by morphological openings using disks as structuring elements was computed. Figure 3 shows the granulometric curves of the three regions. One observes that the granulometries of regions A and S are similar. Particularly, regions A and S contain larger nodules than region L, whereas region L has a greater percentage of intermediate size nodules. According to these results and those of the volume fractions of table 1 it can be assumed that it is not the amount of graphite nodules which plays the main role as lubricant but the size of nodules.

4 Size and Spatial Distributions of Graphite Nodules Based on Connectivity by Dilations

4.1 Multiscale Connectivity

Recently, Braga-Neto and Goutsias [1,2] have investigated the connectivity notion in a multiscale framework. Between the different proposals, the authors introduced the connectivity measure as an approach to quantify the degree of connectivity of an object and showed that this approach is equivalent to connectivity pyramids. A particular and interesting case of connectivity pyramid or connectivity measure is that generated by dilates. Let $E = Z^2$ and a lattice $\sum = \{-R\ , -(R-1), ..., 0\}$ where R a positive integer. Consider a structuring element $\sigma B \in \mathcal{C}$ and containing the origin. Then, equation 4 define a connectivity pyramid.

$$\mathcal{C}(\sigma) = \{X \in \mathcal{P}(E) : \delta_{|\sigma|B}(X) \in \mathcal{C}\} \tag{4}$$

with $\sigma = -R, -(R-1), ..., 0$.

Thus, using these concepts, the set X in Fig. 2(a) is disconnected for $\sigma < 32$ and connected for $\sigma \geq 32$. However, if the set X is classified in $\mathcal{C}(32)$, nothing can be said about the connectivity for the set at dilation 11 (connected components in $\mathcal{C}(11)$. This is a main drawback since, for example, even though there exists a great difference between the sets in Figs. 4 (a) and (b), both are classified as connected components in the class $\mathcal{C}(21)$. Particularly, considering the spatial

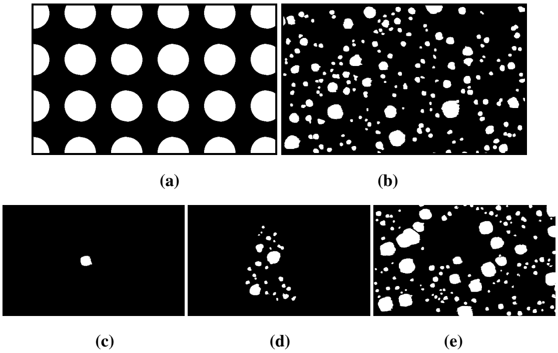


Fig. 4. Connected components. (a), (b) Both sets are connected components of $\mathcal{C}^{\delta_{21}}$, (c), (d), (e) Mean clusters of $\mathcal{C}^{\delta_5}$, $\mathcal{C}^{\delta_{10}}$, $\mathcal{C}^{\delta_{15}}$.

distribution of particles, it is observed that the set in Fig. 4(a) is composed of an array of equidistance particles, while the one in Fig. 4(b) contains a set of different distances between particles. Then, the idea to characterize connectivity consists in classifying a set according to the evolution of connected components when a sequence of dilations is applied. For instance, in Fig. 2, before the set X becomes a connected component of the class $\mathcal{C}(32)$, it has two connected component of the class $\mathcal{C}(11)$. A good representant of these connected components is any of them since both have the same area. In our practical problem, one takes at each dilation σ the mean connected component at this connectivity $\mathcal{C}(\sigma)$. The images in Figs. 4(c)-(e) show some class representant at dilations 5, 10 and 15 (clusters with area value closest to the mean value).

4.2 Axisymmetric Unit Cells Model

Nodules spatial distribution in spheroid cast irons plays a fundamental role not only for studying the stress distribution but also to understand other mechanical properties. Then, the main goal in this section is focused on the use of a well-known model, frequently applied to study mechanical properties in ductile cast iron, in order to understand the wear behavior. In fact, the nodular cast iron material system well suits to the scheme of repetitive unit cells (see Fig. 5(a)), because graphite nodules embedded in the ferrous matrix are quite uniformly distributed. Thus, the nodular cast iron microstructure is now idealized as a periodic array of alternating, equally spaced, graphite nodules. This kind of approach is called axisymmetric unit cells model. In order to characterize the spatial distribution of nodules, this model is used as a reference (an ideal spatial distribution model) to which other spatial distributions are compared. To do that, it is assumed that more dispersed is the distribution of distances the worse the spatial distribution is. The images in Figs. 5(b) and (c) were obtained from the axisymmetric unit cells model in Fig. 5(a) by introducing between the objects two and four distances, respectively. Therefore, these images (Figs. 5(b) and (c))

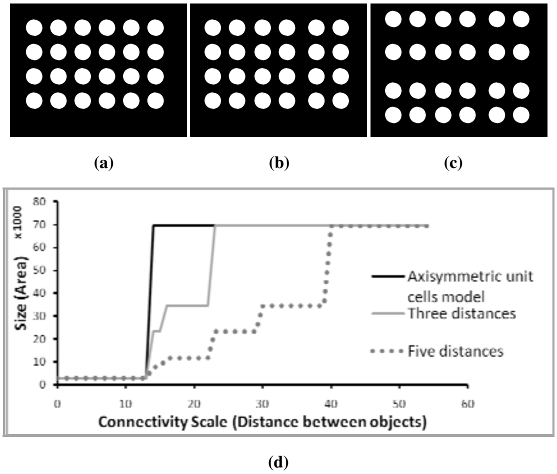


Fig. 5. (a) Axisymmetric unit cells model, (b), (c)Images with three and five distances between the objects, respectively, (d) Curves of mean clusters of images (a), (b),(c)

have three and five distances between the objects. Now, in order to compare both structures of Figs. 5(b) and (c) with the axisymmetric unit cells model in Fig. 5(a), the mean cluster area is computed. The following procedure is made. A sequence of dilations from size 0 to size M is carried out, and at each dilation, the area of the mean cluster is computed. At this point is important to note that for a dilation size smaller than 15 the mean cluster in Figs. 5(a)-(c) is the same and corresponds to the area of the elementary particle (disk), whereas for a dilation size greater than 41 the mean cluster value is also the same, but in this case, it corresponds to the complete area of the objects. In Fig. 5(d) the curves of the mean cluster values for the three images are illustrated. Observe that the curve in gray color, corresponding to the image in Fig. 5(b), is closer to the curve in black color of the axisymmetric unit cells model distribution image in Fig. 5(a).

This model was applied to the region A, L and S to determine the spatial distribution behavior of graphite nodules. Curves of mean clusters of regions A and S, in Fig. 6, are similar and different from the curve of the region L. However, all curves have some similarities, particularly, they are comparable inside two intervals; between 0 and 14 μm , and after 33.6 μm. This means, on the one hand, that the smallest nodules distance of three regions is more or less the same (14 μm) and the mean cluster size corresponds approximately to the nodule mean size. On the other hand, for the largest distance (33.6 μm), the mean cluster of three regions is also very similar and the mean cluster size, in this case, is related to the volume fraction. As it was already shown in Section 3, these values, nodule mean size and volume fraction, are practically the same for the three regions. Therefore, three regions can be established; a) one zone before distances smaller than 14 μm where the mean cluster keeps up a correspondence with the mean nodule value, b) a region between 14 and 33.6 μm where the mean cluster value

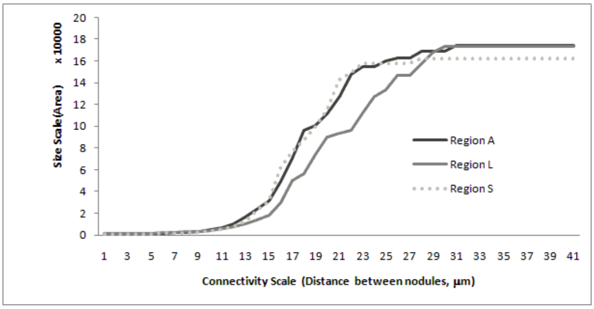


Fig. 6. (a), (b), (c) and (d) Clusters at dilation 0, 5, 8 and 11, (e) and (f) Curves of mean clusters of regions A, L and S

increases approximately in a linear way, and c) a region after 33.6 μm where the stability is reached. The behavior of three regions gives us some information of solidification rate. First, there is an homogeneity between thin and thick regions of the mechanical part since all samples have, more or less, the same minimum distance between nodules. On the other hand, the linear behavior in the second zone of the curves also describes an homogeneity behavior since the mean cluster size increases uniformly. The linear approximations in this zone for the regions A and S are $10.3e3 \cdot |\sigma| - 160.7e3$ and $10.5e3 \cdot |\sigma| - 161.3e3$, respectively, whereas that of the region L is $8.3e3 \cdot |\sigma| - 137.4e3$. Region A and S have greater slopes than L as it is the case in the wear behavior (see Table 1) where the region A and S present a greater resistance to the wear than region L. Thus, the nodules cluster, which integrates size and nodules distance, takes the main role in the wear behavior. In fact, since curves of regions A and S have greater slopes than the curve of region L, then these regions contain greater clusters than region L.

5 Conclusion

In the present work, spatial and size distributions have been studied in order to characterize the microstructure of graphite. We have shown that the nodularity, volume fraction and mean size do no play a main role in the wear behavior. Only the nodule density correlates with the wear percentage, but it does not strictly explains the lubricating effect of nodules. Contrarily, spatial and size distributions gave us interesting information about the tribological behavior of nodules in a ductile cast iron. Whereas a granulometry study has shown that the size of nodules is an important parameter to understand the lubricanting behavior, a study of the spatial homogeneity of nodules better explains this behavior. Indeed, clusters generated by dilations, that takes into account both concepts, size and spatial distributions, seem to be the main geometrical structures to better understand the wear resistance.

Acknowledgments. The authors Ana M. Herrera-Navarro and Hayde Peregrina-Barreto thank the government agency CONACyT for the financial support. The author I. Terol would like to thank Diego Rodrigo and Darío T.G. for their great encouragement. This work was funded by the government agency CONACyT (133697), Mexico.

References

1. Braga-Neto, U., Goutsias, J.: A Multiscale Approach to Connectivity. Computer Vision and Image Understanding 89, 70–107 (2003)
2. Braga-Neto, U.: Multiscale Connected Operators. J. of Math Imaging and Vis. 22, 199–216 (2005)
3. Dommarco, R.C., Jaureguiberry, A.J., Sikora, J.A.: Rolling contact fatigue resistance of ductile iron with different nodules counts and matrix microstructures. Wear 261, 172–179 (2006)
4. Heijmans, H.: Morphological image operators. Academic Press, USA (1994)
5. Heijmans, H.: Connected morphological operators for binary images. Computer Vision and Image Understanding 73(1), 99–120 (1999)
6. Imasogie, B.I., Wendt, U.: Characterization of graphite particle shape in spheroidal graphite iron using a computer-based image analyzer. Journal of Minerals and Materials Characterization and Engineering 3, 1–12 (2004)
7. Karl-Fredrik, N., Vratko, V.: Analysis of ductile cast iron tensile ductility variation to casting defects and material microstructure. Materials Science and Engineering A 502, 54–63 (2009)
8. Morales-Hernandez, L.A., Terol-Villalobos, I.R., Dominguez-Gonzalez, A., Manriquez-Guerrero, F., Herrera-Ruiz, G.: Spatial distribution and spheroidicity characterization of graphite nodules based on morphological tools. Journal of Materials Processing Technology 210(2), 335–342 (2010)
9. Nabil, F., Aly, A., Moenes, S.: C, Si and Ni as alloying elements to vary carbon equivalent of austenitic ductile cast iron: microstructure and mechanical properties. Materials Science and Engineering A 504, 81–89 (2009)
10. Ronse, R.: Set-theoretical algebraic approaches to connectivity in continuous or digital spaces. J. of Math Imaging and Vis. 8, 41–58 (1998)
11. Ronse, C., Serra, J.: Geodesy and connectivity in lattices. Fundadenta Informaticae 46, 349–395 (2001)
12. Serra, J.: Image Analysis and Mathematical Morphology. Theoretical advances, vol. 2. Academic Press, New York (1988)
13. Serra, J.: Connectivity on complete lattices. J. of Math. Imaging and Vis. 9(3), 231–251 (1998)
14. Serra, J.: Connection for sets and functions. Fundamenta Informaticae 41, 147–186 (2000)
15. Soille, P.: Morphological Image Analysis: Principles and Applications. Springer, Berlin (1999)
16. Sosa, A.D., Echeverría, M.D., Moncada, O.J., Míngolo, N., Sikora, J.A.: Influence of nodule count on residual stresses and distortion in thin wall ductile iron plates of different matrices. Journal of Materials Processing Technology 209, 5545–5551 (2009)
17. Tzafestas, C.S., Maragos, P.: Shape connectivity: multiscale analysis and application to generalized granulometries. J. of Math. Imaging and Vis. 17, 109–129 (2002)

18. Xin, T., Hong, Z., Lu-quan, R., Zhi-hui, Z., Ren-doing, C.: Effects of graphite shape on thermal fatigue resistance of cast iron with biomimetic non-smooth surface. International Journal of Fatigue 31, 668–677 (2009)
19. Wilkinson, M.H.F., Ouzounis, G.K.: Advances in connectivity and connected attribute filters. In: Advances in Imaging and Electron Phisics, ch. 5, pp. 211–275 (2010)

Topic Segmentation: Application of Mathematical Morphology to Textual Data

Sébastien Lefèvre[1] and Vincent Claveau[2]

[1] VALORIA Laboratory, University of South Brittany. Address: VALORIA Lab,
Campus de Tohannic, BP 573, 56017 Vannes Cedex, France
sebastien.lefevre@univ-ubs.fr

[2] IRISA-CNRS. Address: IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France
vincent.claveau@irisa.fr

Abstract. Mathematical Morphology (MM) offers a generic theoretical framework for data processing and analysis. Nevertheless, it remains essentially used in the context of image analysis and processing, and the attempts to use MM on other kinds of data are still quite rare. We believe MM can provide relevant solutions for data analysis and processing in a far broader range of application fields. To illustrate, we focus here on textual data and we show how morphological operators (here the morphological segmentation using watershed transform) may be applied on these data. We thus provide an original MM-based solution to the thematic segmentation problem, which is a typical problem in the fields of natural language processing and information retrieval (IR).

More precisely, we consider here TV broadcasts through their transcription obtained by automatic speech recognition. To perform topic segmentation, we compute the similarity between successive segments using a technique called *vectorization* which has recently been introduced in the IR field. We then apply a gradient operator to build a topographic surface to be segmented using the watershed transform. This new topic segmentation technique is evaluated on two corpora of TV broadcasts on which it outperforms other existing approaches. Despite using very common morphological operators (i.e., the standard Watershed Transform), we thus show the potential interest of MM to be applied on non-image data.

1 Introduction

Mathematical Morphology (MM) has led to many successes in image analysis and processing. But its theoretical framework is much more general, and one can wonder why MM stays almost unknown in other fields, while it is expected to provide robust and efficient data analysis tools. In this paper, we address this issue and show that MM may provide very relevant solutions to problems encountered with non-image data. More precisely, we focus on topic segmentation which is a common problem of (Multimedia) Information Retrieval (IR) and Natural Language Processing.

P. Soille, M. Pesaresi, and G.K. Ouzounis (Eds.): ISMM 2011, LNCS 6671, pp. 472–481, 2011.

Topic segmentation is of high interest in Multimedia IR. Indeed, it is needed to perform automatic structuring of TV streams, a keystone for every processing of such streams, which is still done manually in national archive agencies like the French INA. A way to obtain this structuration is to first transcribe the audio tracks of the TV streams into textual data, and then perform the topic segmentation from textual data to split the streams into semantic units (e.g., reports).

In this paper we address the problem of topic segmentation of textual data in this applicative framework using mathematical morphology. To do so, we show that topic segmentation and image segmentation have common characteristics (Sec. 2). From this observation we build a topic segmentation method based on the watershed transform. Moreover, we suggest to build the topographic surface from which the watershed lines are identified using a gradient computation method adapted to the problem under consideration, and thus consider here a vectorization-based gradient approach inspired from recent advances in IR (Sec. 3). Experiments performed on two TV broadcast corpora are presented and discussed (Sec. 4). Finally, Sec. 5 concludes this work and provides future research directions.

2 From Image to Text: Links between Morphological and Topic Segmentation

2.1 Morphological Segmentation

Mathematical morphology is both a rich theoretical framework and a complete toolbox of efficient and robust tools for solving image analysis and processing problems. Among these problems, image segmentation aims to split the input image into a set of uniform regions given a predefined uniformity criterion (intensity or colour, texture, etc.) and is a preprocessing step required by many computer vision tasks. The most famous morphological method for image segmentation is certainly the watershed transform, even if connective segmentation has gathered great interest recently.

We recall very briefly the principle of watershed-based segmentation [10]. The image I to be segmented is first represented as a topographic surface. Watershed lines identified on this surface are then associated to region frontiers resulting from the segmentation process. This relatively simple principle led to various paradigms, and we consider here the flooding approach. It simulates the progressive flooding of the surface starting from its local minima, and builds some dams to avoid merging water from two different catchment basins. At the end of the process, dams correspond to the watershed lines or, in other words, to the region frontiers.

Most often, this approach is not directly applied on the image I to be segmented, since it would then seek for frontiers of high intensity (watershed lines) separating areas of lower intensity (catchment basins). Before applying the segmentation, an image transform is rather performed as a preprocessing in order to highlight values of edge pixels and to lower pixel values in homogeneous areas. Among the transforms which may be involved, an image gradient (noted ∇I hereafter) is usually computed to enhance transition areas (which generally correspond to object frontiers). Various gradient computation methods exist; the most famous ones rely on a convolution with a weighted local mask to measure the dissimilarity within the neighbourhood of each pixel. The choice of the transform to be applied before the segmentation is of high importance, since it will directly influence the segmentation result produced by the watershed method. Indeed, gradient computation methods are often very sensitive to noise, and they often produce many local minima. Since each local minimum is associated to a new catchment basin, and thus to a new region in the image, watershed segmentation most often faces the problem of oversegmentation.

In order to reduce oversegmentation, several strategies may be considered: defining a robust gradient, including some oversegmentation reduction steps in the process (e.g., by merging basins or regions), or setting the predefined number of regions with markers which define the initial catchment basins [7]. These strategies may be supervised or not, respectively leading to user/knowledge-based and automatic methods. A supervised approach for morphological segmentation has been recently introduced in [3]. It relies on a fuzzy classification of the input multipsectral image, and suggests among other contributions to compute the gradient on class membership values associated to each pixel rather than original multispectral pixel values in the input image. Regions built by this method are thus composed of neighbouring pixels which share the same similarities to user-predefined classes, but not necessarily the same values in the input image. This method has been shown to reduce the oversegmentation phenomenon by considering a more robust data representation space. Later in this paper, we will inspire from this principle to transpose the morphological segmentation to the problem of topic segmentation.

2.2 From Image to Text

The analogy between image and text segmentation can be drawn very simply. The pixel is the base element in the image and is described by its greylevel or color/multispectral values. Its equivalent in texts is the sentence (or sometimes the paragraph) which is described by the words it contains.

In our framework of video segmentation, our texts are obtained from automatic transcription. These transcriptions are not composed of sentences but of utterances (sequence of words spoken between two breaths or silences) identified by a timestamp. These utterances are the minimal units of the text (i.e., they are equivalent to image pixels) and topic breaks will be sought between them.

Besides, our texts are flows of utterances. They are then represented as 1-D signal, while images are most often 2- or 3- dimensional. However, nothing prevents the watershed technique to be applied on a single dimension. Thus our approach relies on a gradient computed on the sequence of utterances, and topic breaks are identified using the watershed transform. Gradient computation, which is a key step of the segmentation process, is detailled in Sec. 3. The watershed technique used here is the standard one described previously. We have only included a gradient smoothing step to remove irrelevant local minima.

3 Gradient Computation Using Vectorization

3.1 Vectorization Principles

Vectorization is an embedding technique which aims to project any similarity computation between two documents (or one document and one request in the context of IR) in a vectorial space. It has been introduced and experimented in a standard IR scenario [2] where it has shown to provide both a low complexity and accurate results. We recall here its main principles.

Its principle is relatively simple. For each document of the considered collection, it consists in computing with an initial similarity measure (eg. standard similarity measure used in IR), whatever it is, some proximity scores with m pivot-documents. These m scores are then gathered into a m-dimensional vector representing the document (*cf.* Fig. 1).

Comparing two documents (or a document and a request) can then be performed in a very standard way in this vectorial space (e.g., using a L_2 distance). Many algorithms are available to compute or approximate very efficiently such distances.

More formally, we note $\text{Vect}(D, \mathcal{P}, Sim)$ the vector representing the document D built from the initial similarity measure Sim on pivot-documents $\mathcal{P}$. For instance, $\text{Vect}(D, [P_1, P_2, P_3], \text{TF.IDF/cosine})$ is a 3-dimensional vector; its first component is the similarity score between the document D and the pivot-document P_1 returned by a system using TF.IDF representation associated to the cosine distance measure (which corresponds to a very standard way to compute similarities in the IR field; TF and IDF respectively stand for Term Frequency and Inverse Document Frequency [8, for details]), and so on for the next components.

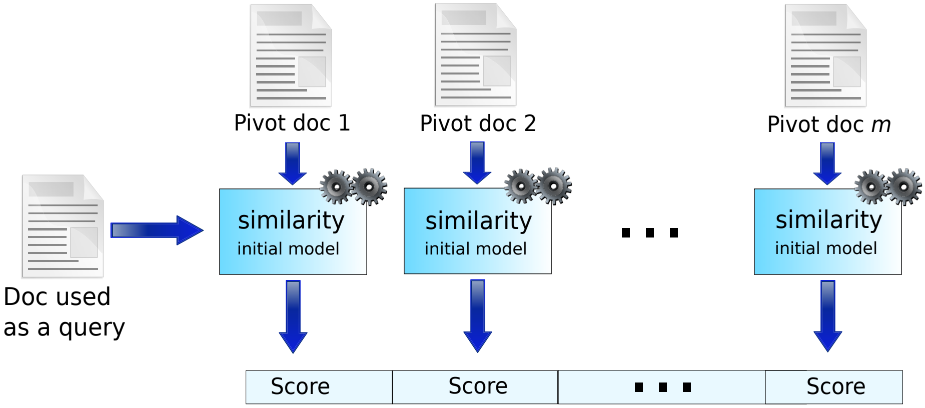


Fig. 1. Vector design from pivot-documents

3.2 Properties

It is important to notice that vectorization results in a change of the representation space, contrary to existing works consisting rather of a dimension reduction or a distance approximation (e.g., [1]). This space transform offers several nice properties which will be discussed here.

The first interest of this embedding is to reduce complexity when the initial similarity computation may be computationally expensive (e.g., some graph comparison computations used in complex IR systems). In a IR context, vectors associated with each document may be built offline, and when a request has to be processed, we only need to compute its similarity with the m pivot-documents rather than to do it with all documents in the collection. This property is nevertheless not useful in the context of a segmentation task.

The second nice property comes from the fact that two documents will be considered as similar if they are similar to the same pivot-documents. This indirect comparison, or second-order affinity, let us compare two textual documents which do not share any common word. This property will be helpful in our segmentation task. Indeed, it will solve the problem brought by the lack of repetition between utterances. This problem is particularly noticeable when the segments to be compared are of short duration (i.e., they will contain less words, and thus will share only a few words in common in the best case, and no common words in the worst case).

3.3 Usage

A gradient is computed between each utterance. In other words, we compute the similarity using the vectorization principle between previous and next utterances. Let us note that we do not compare only the previous to the next utterance, but we also consider the n previous ones vs. the n next ones (similarly to TEXTTILING, a common approach for topic segmentation).

In experiments described in the following section, the initial similarity measure used in the vectorization process is a L_2 distance associated with a weighting of utterances by $\sqrt{TF}$. It means that we first represent each breath group by a sparse vector in which each dimension represent a word; the value for this dimension is the square root of the number of occurrences of the word in the breath group. The same is done for the pivot document. The distance between the breath group vector and the pivot vector is computed with a L_2 distance; the resulting value forms one of the dimension of the new vector.

Similarly to some image gradient computation methods (e.g., Sobel), we give more importance to close utterances and less importance to utterances which are far to the candidate edge. This is ensured through a simple convolution with a kernel (e.g., Gaussian kernel). Let us notice that the way the convolution is applied depends on the way the documents are represented in the initial model of similarity computation. With the vectorial representation used in our experiments, this convolution is simply taken into account: when computing $\sqrt{TF}$, the occurrence of a word counts for one in the breath group which is the closest from the candidate edge, but counts for less when considering an occurrence from a breath group further of the candidate edge. In practice, a linear penalty is applied. From now we will write $\mathcal{C}_{prev}(i)$ (respectively $\mathcal{C}_{next}(i)$) the result of the convolution operator applied on utterance i and those which are preceding (respectively following) it.

Formally, the gradient is thus defined by:

$$\nabla(i) = L_2(\mathrm{Vect}(\mathcal{C}_{prev}(i-1), \mathcal{P}, \sqrt{TF}/L_2), \mathrm{Vect}(\mathcal{C}_{next}(i), \mathcal{P}, \sqrt{TF}/L_2))$$

Pivot-documents we are using are simply sequences of utterances built from random splits of the considered broadcast. While each utterance is associated to a vectorial representation, we can observe than the gradient in each utterance results in a scalar value. Thus there is no need to use multivariate morphology and to adapt the watershed algorithm.

Fig. 2 shows an example of a vectorization-based gradient computed on a sample of one of our corpora (see below). We can observe that the signal contains local minima and thus needs a smoothing step. As indicated previously, we involve a smoothing step to remove such minima and help the watershed segmentation process. Resulting segmentation is provided in Fig. 3. The considered sample contains 4 segments (ground truth is shown in green full line; detected edges by our system using watershed transform are given in dotted lines). Utterances are represented by their starting time. For a given time index, the higher the smoothed gradient is, the more important the dissimilarity between previous and next groups is. In other words, significant local maxima of gradient values indicate a topic break. Nevertheless, local maxima are not sufficient to identify topic breaks. Indeed, we cannot make any assumption regarding the segment length. Extraction of local maxima by signal analysis with a sliding window is then inappropriate. Moreover, such an approach would have lack of robustness.

4 Experiments

4.1 Experimental Data

Our experiments are performed on two French TV broadcast corpora for which the topic segmentation is of high interest. The first corpus is a set of 60 TV news of the France 2 channel (called *News* further). Each of these sample has been broadcasted in the beginning of 2007 and is 40 minutes long. The second corpus is made from TV reports: 12 samples of *Envoyé spécial* (2008, 2 hours long each), and 16 *Sept à huit* (2008, 1 hour long each). This corpus is called *Reports* in the following experiments.

These corpora [4] have different properties in terms of number and duration of topic segments. Thus, it allows us to evaluate robustness of topic segmentation methods. The *News* corpus contains 1180 segments while the *Reports* corpus only contains 140 segments.

The reference segmentation (i.e., ground truth) has been independently built by a user who was not involved in the design of a topic segmentation system.

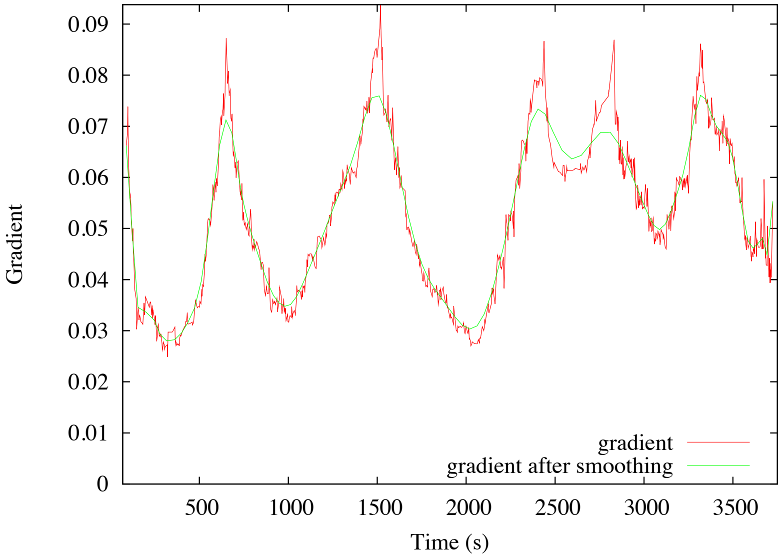


Fig. 2. Effect of the filtering step: gradient vs. starting time of utterances

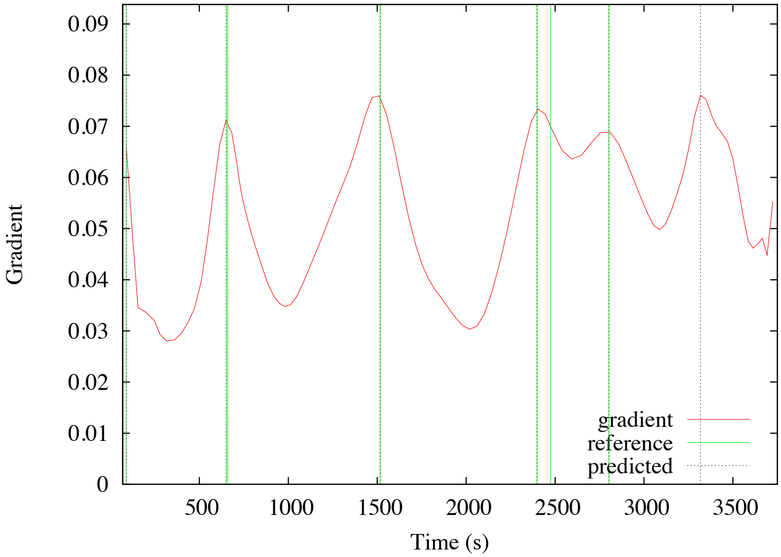


Fig. 3. Illustration of segmentation process: filtered gradient vs. starting time of utterances

Since there is no consensus on the topic definition in the IR or NLP fields, it has been considered here that a topic change occurs for each report change. Despite this assumption being not always valid (in particular in the *News* corpus in which several successive reports may be considered as related to the same topic), it is relevant since it corresponds to an actual and well-defined applicative need.

4.2 Data Preprocessing

Audio tracks of these two corpora have been automatically transcribed using the speech recognition system IRENE [6]. This system has been initially designed for transcribing radio broadcasts for which it produces a Word Error Rate of about 20%. In our context of transcribing TV broadcasts, it is very probably higher due to the more noisy environment we have to face. Transcriptions are finally part-of-speech tagged using TreeTagger[1], and only names, verbs, and adjectives are kept and stemmed.

4.3 Results

Recall, precision, and F1-measure are used as quality measures to evaluate our proposed method. We consider that a segment edge is correct as soon as it is located in the close neighbourhood (less than 10 seconds) of a reference frontier. In order to show the relevance of our contribution, we compare the results obtained by our MM-based method to those (when available) produced by several existing systems on the same corpora: the system from Utiyama and Isahara [9] relying on a Hidden Markov Model (we use the implementation from [4]), and the best results obtained from the system of [4]. We also provide results obtained by a self-implementation of TEXT-TILING [5] in which we use the same data preprocessing and the same watershed-based segmentation framework. The only difference is the way the gradient is computed (i.e., without vectorization), which can be here written:

$$\nabla(i) = \mathit{cosine}(\text{TF-IDF}(\mathcal{C}_{prev}(i-1)), \text{TF-IDF}(\mathcal{C}_{next}(i)))$$

The TEXT-TILING approach aims to find topic breaks where lexical coherence between previous and next text blocks is linked to a significant local minimum. That is why we have derived a watershed-based implementation by considering the inverse of the lexical coherence measure as the topographic surface to be used in the watershed process.

Table 1 shows results obtained on both corpora. In both cases, we can observe that our system provides better performances than existing systems. In order to better understand the interest of using MM for topic segmentation, we compare more deeply the approach introduced in this paper and our own implementation of TEXT-TILING. Both are based on the Watershed transform but they differ on the gradient computation method. We rely here on a vectorization technique

[1] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger

Table 1. Performances of topic segmentation systems on *News* and *Reports* corpora

method	*News* corpus			*Reports* corpus		
	Precision	Recall	F1-Measure	Precision	Recall	F1-Measure
Utiyama [9]	-	-	59.44	-	-	51.09
Guinaudeau [4]	-	-	61.41	-	-	62.92
TEXT-TILING [5]	44.17	41.97	43.04	59.32	60.93	60.12
proposed	**67.47**	**61.6**	**64.4**	**77.38**	**69.65**	**73.31**

rather than a cosine measure combined with a TF-IDF representation (a common approach in IR). The superiority of our approach is particularly observable on the *News* corpus, since this corpus contain very short segments, thus making the direct computation of the gradient as done in TEXT-TILING unreliable. In other words, the straight application of morphological operators may be of lower interest. It is much more relevant to adapt the morphological process to the data under consideration, e.g. here to use an appropriate gradient computation method.

5 Conclusion

In this paper, we aim to show that Mathematical Morphology can be successfully applied on non-image data. To do so, we consider the topic segmentation problem faced in the fields of information retrieval and natural language processing. We show the parallel which can be driven between topic and image segmentation. From this parallel we were able to introduce a new topic segmentation method based on morphological segmentation using the watershed transform. The results are appealing and strongly suggest that Mathematical Morphology would benefit to many fields and not only to image analysis and processing.

Moreover, we have included in this approach a vectorization-based gradient computation method. The experiments we have made lead to the expected conclusion that Mathematical Morphology should not be straightly applied to textual data. Indeed, it is more relevant to adapt some steps of the morphological data processing scheme to the data under consideration. More precisely, the vectorization technique used here is of great help to face the lack of repetitions between utterances, which is an important problem when topic segments are quite short. Applying such a gradient on the input signal, using similarity to given samples (here the pivot-documents), is not new in the field of watershed-based image segmentation. In a previous work [3], we have already suggested to build the topographic surface through a gradient computation made on probability maps obtained from a supervised image classification. The regions produced by this method are then composed of neighbouring pixels sharing the same similarities to predefined classes rather than similarities in the initial multispectral image space. A full adaptation of this strategy to textual data will result in defining highly reliable pivot-documents (which would be more discriminative than the random ones used in this article by the vectorization technique).

As we briefly recalled in Sec. 2, there exist other techniques leading to a better segmentation result. In particular, future work will consider the intelligent morphological segmentation paradigm. In complement to the definition of a relevant image transform to build the topographic surface, the markers could also be of great interest. Besides, marker-based watershed segmentation of textual data may be a way to involve the user in the process if required. Moreover, we also consider hierarchical morphological segmentation schemes to build a multiscale topic segmentation result. This could be of great interest for the Multimedia IR community.

References

1. Abraham, I., Bartal, Y., Neiman, O.: Advances in metric embedding theory. In: Proc. of Symposium on Theory of Computing, Seattle, USA (2006)
2. Claveau, V., Tavenard, R., Amsaleg, L.: Vectorisation des processus d'appariement document-requête. In: 7e Conférence en Recherche d'informations et Applications, CORIA 2010, Sousse, Tunisie, pp. 313–324 (March 2010)
3. Derivaux, S., Forestier, G., Wemmert, C., Lefèvre, S.: Supervised segmentation using machine learning and evolutionnary computation. Pattern Recognition Letters 31(15), 2364–2374 (2010)
4. Guinaudeau, C., Gravier, G., Sébillot, P.: Utilisation de relations sémantiques pour améliorer la segmentation thématique de documents télévisuels. In: Actes de la Conférence Traitement automatique des Langues, Montréal, Canada (2010)
5. Hearst, M.: Text-tiling: segmenting text into multi-paragraph subtopic passages. Computational Linguistics 23(1), 33–64 (1997)
6. Huet, S., Gravier, G., Sébillot, P.: Morpho-syntactic post-processing with n-best lists for improved french automatic speech recognition. Computer Speech and Language 24(4), 663–684 (2010)
7. Rivest, J., Beucher, S., Delhomme, J.: Marker-controlled segmentation: an application to electrical borehole imaging. Journal of Electronic Imaging 1(2), 136–142 (1992)
8. Salton, G.: A Theory of Indexing. Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphy (1975)
9. Utiyama, M., Isahara, H.: A statistical model for domain-independent text segmentation. In: Proceedings of the 9th Conference of the ACL (2001)
10. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(6), 583–598 (1991)

Author Index