Cris L. Luengo Hendriks
Gunilla Borgefors
Robin Strand (Eds.)

LNCS 7883

Mathematical Morphology and Its Applications to Signal and Image Processing

11th International Symposium, ISMM 2013
Uppsala, Sweden, May 2013
Proceedings

Springer

Lecture Notes in Computer Science 7883

Cris L. Luengo Hendriks
Gunilla Borgefors Robin Strand (Eds.)

Mathematical Morphology and Its Applications to Signal and Image Processing

11th International Symposium, ISMM 2013
Uppsala, Sweden, May 27-29, 2013
Proceedings

Springer

Volume Editors

Cris L. Luengo Hendriks
Gunilla Borgefors
Robin Strand

Swedish University of Agricultural Sciences and Uppsala University
Centre for Image Analysis
Box 337
751 05 Uppsala, Sweden

E-mails:
cris@cb.uu.se
gunilla@cb.uu.se
robin@cb.uu.se

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-38293-2 e-ISBN 978-3-642-38294-9
DOI 10.1007/978-3-642-38294-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013937542

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

The volume you hold in your hand collects the papers accepted for presentation at the 11th International Symposium on Mathematical Morphology (ISMM 2013), held in Uppsala, Sweden, during May 27–29, 2013. ISMM has been held approximately every two years since 1993, when the series was initiated in Barcelona. The ten preceding editions of this conference were very successful, and the series has established itself as the main scientific event in the field.

We received 52 high-quality papers, each of which was sent to at least three Program Committee members for review. Based on 153 detailed reviews, we accepted 33 papers and conditionally accepted another eight. These eight papers were accepted after substantial revision by the authors in response to reviewer concerns. We decided to conditionally accept papers that needed important changes because each of us has listened to the presentation of a paper we reviewed, only to realize that the authors did nothing with our comments. In this case, however, and to our delight, even papers we accepted without conditions were in most cases extensively rewritten in response to reviewer comments.

In addition to the 41 reviewed papers, this volume contains three papers authored by our invited speakers:

- "Adaptive Morphologic Regularizations for Inverse Problems," by Bhabatosh Chanda (Indian Statistical Institute) with P. Purkait
- "The Laplace-Beltrami Operator: A Ubiquitous Tool for Image and Shape Processing," by Ron Kimmel (Technion–Israel Institute of Technology) with A. Wetzler, Y. Aflalo, and A. Dubrovina
- "Geography, Mathematics and Mathematical Morphology," by Christine Voiron-Canicio (University of Nice–Sophia Antipolis).

We would like to thank everyone involved in putting together this volume and the conference: the authors for providing the scientific content; the Program Committee and the additional reviewers for thorough reviews and detailed comments; the Steering Committee for giving us the opportunity to organize this conference and supporting us throughout the process; Springer for doing most of the work involved in putting this volume together; our two universities for financial support; The Swedish Research Council for sponsoring two of our invited speakers; the International Association for Pattern Recognition and Centre for Interdisciplinary Mathematics (Uppsala University) for sponsoring the third invited speaker; and the City of Uppsala for sponsoring the conference dinner at Östgöta nation. Finally, we would like to acknowledge EasyChair, the online conference management system we used free of charge to handle the paper submission and review process.

March 2013

C. Luengo
G. Borgefors
R. Strand
C. Kiselman
V. Ćurić

Organization

ISMM 2013 was organized by the Centre for Image Analysis, a collaboration between Uppsala University and the Swedish University of Agricultural Sciences.

Organizing Committee

Gunilla Borgefors	General Chair
Christer Kiselman	Invited Speakers' Chair and Adviser
Cris L. Luengo Hendriks	Program Chair
Robin Strand	Local Organization Chair
Vlada Ćurić	Assistant

Steering Committee

Jesús Angulo	Mines ParisTech, France
Junior Barrera	University of São Paulo, Brazil
Isabelle Bloch	Télécom ParisTech, France
Gunilla Borgefors	Uppsala University, Sweden
Renato Keshet	Hewlett Packard Laboratories, Israel
Ron Kimmel	Technion–Israel Institute of Technology, Israel
Petros Maragos	National Technical University of Athens, Greece
Christian Ronse	University of Strasbourg, France
Philippe Salembier	Technical University of Catalonia, Spain
Dan Schonfeld	University of Illinois at Chicago, USA
Pierre Soille	EC Joint Research Centre, Italy
Hugues Talbot	University of East Paris, France
Michael H.F. Wilkinson	University of Groningen, The Netherlands

Program Committee

Jesús Angulo
Akira Asano
Junior Barrera
Jon Atli Benediktsson
Isabelle Bloch
Gunilla Borgefors
Michael Buckley
Bernhard Burgeth
Bhabatosh Chanda
Jocelyn Chanussot
Jean Cousty
José Crespo
Vladimir Ćurić
Johan Debayle
Etienne Decencière
Adrian Evans
Thierry Geraud
Lionel Gueguen
Allan Hanbury
Marcin Iwanowski
Andrei Jalba
Dominique Jeulin
Renato Keshet
Ron Kimmel
Christer Kiselman
Ullrich Köthe
Sébastien Lefèvre

Roberto Lotufo
Cris L. Luengo Hendriks
Petros Maragos
Beatriz Marcotegui
Petr Matula
Fernand Meyer
Laurent Najman
Georgios Ouzounis
Nicolas Passat
Jos Roerdink
Christian Ronse
Philippe Salembier
Gabriella Sanniti Di Baja
Jean Serra
Behara Seshadri Daya Sagar
Ida-Maria Sintorn
Pierre Soille
Robin Strand
Hugues Talbot
Iván Ramon Terol-Villalobos
Matthew Thurley
Marc Van Droogenbroeck
Michel Westenberg
Michael H.F. Wilkinson

Additional Reviewers

Mauro Dalla Mura
Nicola Falco
Bangalore Ravi Kiran
Prashanth Marpu
Benjamin Perret

Table of Contents

Theory

Trees and Hierarchies

Adaptive Morphology

Colour

Manifolds and Metrics

Filtering

Detectors and Descriptors

Applications

Similarity between Hypergraphs Based on Mathematical Morphology

Isabelle Bloch[1], Alain Bretto[2], and Aurélie Leborgne[3,⋆]

[1] Institut Mines-Telecom, Telecom ParisTech, CNRS LTCI, Paris, France
isabelle.bloch@telecom-paristech.fr

[2] Greyc Cnrs-Umr 6072, Bd Marechal Juin BP 5186, Caen, France
alain.bretto@info.unicaen.fr

[3] Université de Lyon, INSA-Lyon, CNRS LIRIS, France
aurelie.leborgne@liris.cnrs.fr

Abstract. In the framework of structural representations for applications in image understanding, we establish links between similarities, hypergraph theory and mathematical morphology. We propose new similarity measures and pseudo-metrics on lattices of hypergraphs based on morphological operators. New forms of these operators on hypergraphs are introduced as well. Some examples based on various dilations and openings on hypergraphs illustrate the relevance of our approach.

Keywords: Hypergraphs, similarity, pseudo-metric, valuation, mathematical morphology on hypergraphs.

1 Introduction

The notion of similarity plays a very important role in various fields of applied sciences. Classification is an example [6], and other examples such as indexing, retrieval or matching demonstrate the usefulness of the concept of similarity [7], with typical applications in image processing and image understanding. A recent trend in these domains is to rely on structural representations of the information (images for instance). Beyond the classical graph representations, and the associated notion of graph similarity, hypergraphs (in which edges can have any cardinality and are then called hyperedges), introduced in the 1960s [23], have recently proved useful. This concept has developed rapidly and has become both a powerful and well-structured mathematical theory for modeling complex situations. This theory is now widely used in sciences as diverse as chemistry, physics, genetics, computer science, psychology... [23], most of them requiring the notions of comparison and similarity measure. In image applications, most similarity measures rely on features computed locally, or among the vertices of an hyperedge, and therefore do not completely exploit the structure of the hypergraph at this level [10,11,14,15].

[⋆] This work was partially funded by a grant from Institut Mines-Telecom / Telecom ParisTech, and was done during the sabbatical stay of A. Bretto at Telecom ParisTech and during the Master thesis of A. Leborgne at Greyc.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 1–12, 2013.

In this paper, we propose new tools for defining similarity measures and metrics based on mathematical morphology. In order to deal with structured information, mathematical morphology has been developed on graphs [8,17,21,22], triangulated meshes [16], and more recently on simplicial complexes [9] and hypergraphs [2,3], where preliminary notions of dilation-based similarity were introduced. In this paper, we propose to study similarities on lattices and more specifically on lattices of hypergraphs. We define some of them based on valuations on hypergraphs and mathematical morphology operators. They are illustrated on various types of lattices of hypergraphs, by also introducing new morphological operators, showing the interest of the proposed definitions for achieving robustness with respect to small variations of the compared hypergraphs. This paper is organized as follows. In Section 2 we recall some definitions on hypergraphs and lattices of hypergraphs on which morphological operators are defined. In Section 3, we show some general results on similarities, valuations and pseudo-metrics. Similarity and pseudo-metrics based on mathematical morphology are then defined in Section 4, with a number of illustrative examples.[1]

2 Background and Notations

Basic Concepts on Hypergraphs [5]. A *hypergraph* H denoted by $H = (V, E = (e_i)_{i \in I})$ on a finite set V is a family (which can be a multi-set) $(e_i)_{i \in I}$, (where I is a finite set of indices) of subsets of V called *hyperedges.* Let $(e_j)_{j \in \{1,2,\dots l\}}$ be a sub-family of hyperedges of E. The set of vertices belonging to these hyperedges is denoted by $v(\cup_{j \in \{1,2,\dots l\}} e_j)$, and $v(e)$ denotes the set of vertices forming the hyperedge e. If $\cup_{i \in I} v(e_i) = V$, the hypergraph is without *isolated vertex* (a vertex x is isolated if $x \in V \setminus \cup_{i \in I} v(e_i)$). The set of isolated vertices is denoted by $V_{\setminus E}$. By definition the *empty hypergraph* is the hypergraph $H_\emptyset$ such that $V = \emptyset$ and $E = \emptyset$. A hypergraph is called *simple* if $\forall (i,j) \in I^2, v(e_i) \subseteq v(e_j) \Rightarrow i = j$. The *incidence graph* of a hypergraph $H = (V, E)$ is a bipartite graph $IG(H)$ with a vertex set $S = V \sqcup E$ (where $\sqcup$ stands for the disjoint union), and where $x \in V$ and $e \in E$ are adjacent if and only if $x \in v(e)$. Conversely, to each bipartite graph $\Gamma = (V_1 \sqcup V_2, A)$, we can associate two hypergraphs: a hypergraph $H = (V, E)$, where $V_1 = V$ and $V_2 = E$ and its dual $H^* = (V^*, E^*)$, where $V_1 = E^*$ and $V_2 = V^*$. This notion is useful to prove some results in Section 3.

Mathematical Morphology on Hypergraphs. In [3], we introduced mathematical morphology on hypergraphs. The first step was to define complete lattices on hypergraphs. Then the whole algebraic apparatus of mathematical morphology applies [4,12,13,18,19] and is not recalled here.

We denote the universe of hypergraphs by $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ the set of vertices (that we assume to be finite) and $\mathcal{E}$ the set of hyperedges. The powersets of $\mathcal{V}$

[1] Proofs are omitted because of lack of space.

and $\mathcal{E}$ are denoted by $\mathcal{P}(\mathcal{V})$ and $\mathcal{P}(\mathcal{E})$, respectively. We denote a hypergraph by $H = (V, E)$ with $V \subseteq \mathcal{V}$ and $E \subseteq \mathcal{E}$. As developed in [3], several complete lattices can be built on $(\mathcal{V}, \mathcal{E})$. Let us denote by $(\mathcal{T}, \preceq)$ any of these lattices. We denote by $\wedge$ and $\vee$ the infimum and the supremum, respectively. The least element is denoted by $0_\mathcal{T}$ and the largest element by $1_\mathcal{T}$. Here we will use three examples of complete lattices: $\mathcal{T}_1 = (\mathcal{P}(\mathcal{V}), \subseteq)$, $\mathcal{T}_2 = (\mathcal{P}(\mathcal{E}), \subseteq)$ (which are simply lattices over the power set of vertices and the power set of edges, respectively), and $\mathcal{T}_3 = (\{H\}, \preceq)$ where $\{H = (V, E)\}$ denotes a set of hypergraphs defined on $(\mathcal{V}, \mathcal{E})$ such that $\forall e \in E, v(e) \subseteq V$, and the partial ordering is defined as $\forall (H_1 = (V_1, E_1), H_2 = (V_2, E_2)) \in \mathcal{T}_3^2, H_1 \preceq H_2 \Leftrightarrow V_1 \subseteq V_2 \textit{ and } E_1 \subseteq E_2$ [3]. As shown in [3], we have $H_1 \wedge H_2 = (V_1 \cap V_2, E_1 \cap E_2)$ and $H_1 \vee H_2 = (V_1 \cup V_2, E_1 \cup E_2)$. Examples of dilations on these lattices can be found in [3]. In Section 4, we provide further examples, along with the adjoint erosions, as well as examples of openings.

3 Similarity, Valuation and Pseudo-Metric

3.1 Similarity and Pseudo-metric

A similarity on a set $\mathcal{T}$ is defined as a function from $\mathcal{T} \times \mathcal{T}$ into $[0, 1]$ such that $\forall (x, y) \in \mathcal{T}^2, s(x, y) = s(y, x)$ and $s(x, x) = 1$. We will consider in particular the case where $\mathcal{T}$ is a lattice defined on hypergraphs. From a similarity s, a semi-pseudo-metric can be defined as $\forall (x, y) \in \mathcal{T}^2, d(x, y) = 1 - s(x, y)$. If moreover s satisfies $\forall (x, y, z) \in \mathcal{T}^3, s(x, z) + s(z, y) - 1 \leq s(x, y)$, then d is a pseudo-metric[2].

Proposition 1. *Let w be a positive, monotonous (increasing) real-valued function defined on a lattice $\mathcal{T}$.*

(a) If $\forall (x, y, z) \in \mathcal{T}^3, w(x \wedge y) \leq w(x \wedge z) + w(z \wedge y)$ and $w(x \vee y) \geq \max(w(x \vee z), w(z \vee y))$, then the function d_1 defined as $\forall (x, y) \in \mathcal{T}^2, d_1(x, y) = \frac{w(x \wedge y)}{w(x \vee y)}$ if $w(x \vee y) \neq 0$, and 0 otherwise, is a pseudo-metric.

(b) If $\forall (x, y, z) \in \mathcal{T}^3, w(x \wedge y) \geq w(x \wedge z) + w(z \wedge y)$ and $w(x \vee y) \leq \min(w(x \vee z), w(z \vee y))$, then the function d_2 defined as $\forall (x, y) \in \mathcal{T}^2, d_2(x, y) = 1 - \frac{w(x \wedge y)}{w(x \vee y)}$ if $w(x \vee y) \neq 0$, and 0 otherwise, is a pseudo-metric.

Note that the conditions involved in this proposition are quite strong. In particular, they do not hold for simple valuations such as the cardinality on a graded lattice (see Section 3.2).

Proposition 2. *Under condition (b) above $d(x, y) = w(x \vee y) - w(x \wedge y)$ defines a pseudo-metric.*

[2] For a pseudo-metric, we have $d(x, x) = 0$ but we may have $d(x, y) = 0$ for $x \neq y$, and for a semi-metric the triangular inequality does not necessarily hold. So a semi-pseudo-metric satisfies $\forall (x, y) \in \mathcal{T}^2, d(x, y) = d(y, x), d(x, x) = 0$.

3.2 Valuation on a Lattice $(\mathcal{T}, \preceq)$ and Pseudo-metric

Definition 1. *[1] A valuation w on a lattice $(\mathcal{T}, \preceq)$ is defined as a real-valued function such that: $\forall (x,y) \in \mathcal{T}^2, w(x) + w(y) = w(x \wedge y) + w(x \vee y)$. A valuation is increasing if $\forall (x,y) \in \mathcal{T}^2, x \preceq y \Leftrightarrow w(x) \leq w(y)$.*

In the following we consider only increasing valuations. We have then $\forall x \in \mathcal{T}, w(0_{\mathcal{T}}) \leq w(x) \leq w(1_{\mathcal{T}})$, and $\forall (x,y) \in \mathcal{T}^2, w(x \wedge y) \leq w(x \vee y)$. A pseudo-metric can be derived as follows [1].

Theorem 1. *[1] Let w be an increasing valuation on $(\mathcal{T}, \preceq)$. Then d, defined by $\forall (x,y) \in \mathcal{T}^2, d(x,y) = w(x \vee y) - w(x \wedge y)$ is a pseudo-metric. The following inequality also holds: $\forall (a,x,y) \in \mathcal{T}^3, d(a \vee x, a \vee y) + d(a \wedge x, a \wedge y) \leq d(x,y)$.*

Note that this result requires weaker assumptions than the condition (b) used in Proposition 2.

Let us consider the lattice $\mathcal{T}_1 = (\mathcal{P}(\mathcal{V}), \subseteq)$ [3]. The cardinality defines an increasing valuation: $\forall V \subseteq \mathcal{V}, w(V) = |V|$. We have $w(V) = 0 \Leftrightarrow V = \emptyset = 0_{\mathcal{T}}$ and $w(1_{\mathcal{T}}) = |\mathcal{V}|$. In this case, d is a metric (in particular we have $d(V, V') = 0 \Leftrightarrow V = V'$), and can be expressed as: $\forall (V, V') \in \mathcal{P}(\mathcal{V})^2, d(V, V') = |V \cup V'| - |V \cap V'| = |V| + |V'| - 2|V \cap V'| = |V \Delta V'|$.

Proposition 3. *The lattice $\mathcal{T}_3 = (\{H\}, \preceq)$ is modular.*

Note that if the hypergraphs are supposed to be without isolated vertices, the double partial ordering reduces to inclusion between hyperedge sets and $\mathcal{T}_3$ is isomorphic to $\mathcal{T}_2 = (\mathcal{P}(\mathcal{E}), \subseteq)$ and hence modular (this is derived from the fact that this lattice is distributive). Here we consider the more general case where V can contain isolated vertices.

As shown in [1], on any modular lattice, the height function (i.e. assigning to every $x \in \mathcal{T}$ the least upper bound of the lengths of the chains from $0_{\mathcal{T}}$ to x) defines a valuation w, leading to a graded lattice. This valuation is strictly monotonous. An interesting property is that $\forall (x,y) \in \mathcal{T}^2$, if y covers x (i.e. $x \prec y$ and $\nexists z \in \mathcal{T}, x \prec z \prec y$) then $w(y) = w(x) + 1$.

Proposition 4. *On $\mathcal{T}_3 = (\{H\}, \preceq)$, the valuation defined by the height function is equal to: $\forall H = (V, E) \in \mathcal{T}_3, w(H) = |V| + |E|$.*

4 Mathematical Morphology and Similarity between Hypergraphs

4.1 Similarity and Dilation

If A and B are $m \times m$ matrices, we denote by $A \circ B$ their entry-wise product, i.e. the matrix whose $m_{i,j}$ entry is $a_{i,j}b_{i,j}$. It is called the *Schur or Hadamard product* of A and B. It is known that if A and B are positive definite, then so is $A \circ B$.

Theorem 2. *Let S be a set and s a similarity on S such that $s(x) \in [0;1] \cap \mathbb{Q}_+$, for all $x \in S$ (where $\mathbb{Q}_+$ denotes the set of positive rational numbers). Let us assume that s can be written as:*

$$\forall (u_i, u_j) \in S^2, i, j \in \{1, \ldots, m\},\ s(u_i, u_j) = \left(\frac{x_{i,j}}{x_i + x_j - x_{i,j}} \right)$$

with $x_{i,j} = x_{j,i}$, $x_{i,i} = x_i$ and $x_j \geq x_{i,j}$ Then the matrix $M(s) = (s(u_i, u_j))_{i,j \in \{1,\ldots,m\}}$ can be written as the Hadamard product of two matrices A and B verifying the following properties:

1. *the matrix A is a semi-positive definite symmetric Cauchy matrix (i.e. having the following form: $A = (a_{ij})_{i,j} = \left(\frac{1}{x_i + x_j} \right)_{i,j}$; $\quad x_i + y_j \neq 0$);*
2. *the matrix B is a matrix defined by the following process: there is a simple hypergraph $H = (V, E)$ with $|E| = m$ and a dilation from $(\mathcal{P}(E), \subseteq)$ into $(\mathcal{P}(V), \subseteq)$ such that $B = (|\delta(e_i) \cap \delta(e_j)|)_{i,j \in \{1,\ldots,m\}}$.*

From this result it is easy to show the following result.

Corollary 1. *Let S be a set and s a similarity on S defined as above. Let us assume that s can be written as:*

$$\forall (u_i, u_j) \in S^2, i, j \in \{1, \ldots, m\},\ s(u_i, u_j) = \left(\frac{2x_{i,j}}{x_i + x_j} \right)$$

with $x_{i,i} = x_i$. Then the matrix $M(s) = (s(u_i, u_j))_{i,j \in \{1,\ldots,m\}}$ can be written as the Hadamard product of two matrices A and B verifying the following properties:

1. *the matrix A is a Cauchy matrix;*
2. *the matrix B is a matrix defined by the following process: there is a simple hypergraph $H = (V, E)$ with $|E| = m$ and a dilation on E such that $B = (|\delta(e_i) \cap \delta(e_j)|)_{i,j \in \{1,\ldots,m\}}$.*

4.2 Similarity from a Valuation and a Morphological Operator

Let us consider any lattice of hypergraphs $(\mathcal{T}, \preceq)$, an increasing valuation w and a morphological operator ψ defined on this lattice. In this section, we generalize ideas suggested in [3] in the particular case where the lattice was the power set of vertices, w was the cardinality and ψ was a dilation.

Definition 2. *Let $(\mathcal{T}, \preceq)$ be a lattice of hypergraphs, w an increasing valuation on this lattice such that $w(x) = 0$ iff $x = 0_{\mathcal{T}}$, and ψ a morphological operator from $(\mathcal{T}, \preceq)$ into $(\mathcal{T}, \preceq)$ such that $\psi(x) = 0_{\mathcal{T}} \Rightarrow x = 0_{\mathcal{T}}$. We define a real-valued function s as: $\forall (x, y) \in \mathcal{T}^2 \setminus (0_{\mathcal{T}}, 0_{\mathcal{T}}), s(x, y) = \frac{w(\psi(x) \wedge \psi(y))}{w(\psi(x) \vee \psi(y))}$ and $s(0_{\mathcal{T}}, 0_{\mathcal{T}}) = 1$.*

Proposition 5. *The function s introduced in Definition 2 is a similarity.*

In a similar way as in Theorem 1, we introduce a pseudo-metric defined from w and ψ.

Proposition 6. *Let w and ψ defined on $(\mathcal{T}, \preceq)$ as in Definition 2. The real-valued function d_ψ defined as: $\forall (x,y) \in \mathcal{T}^2, d_\psi(x,y) = w(\psi(x) \vee \psi(y)) - w(\psi(x) \wedge \psi(y))$ is a pseudo-metric. In the particular case where $\mathcal{T}$ is the power set of the set of vertices or of hyperedges (with $\preceq$ equal to $\subseteq$), and the valuation is the cardinality, then d_ψ is a metric.*

Note that again this result requires weaker assumptions than the condition (b) used in Proposition 2.

The similarity s and the pseudo-metric d_ψ are linked by the following relation: $\forall (x,y) \in \mathcal{T}^2 \setminus (0_\mathcal{T}, 0_\mathcal{T}), 1 - s(x,y) = \frac{d_\psi(x,y)}{w(\psi(x) \vee \psi(y))}$ and $1 - s(0_\mathcal{T}, 0_\mathcal{T}) = d_\psi(0_\mathcal{T}, 0_\mathcal{T}) = 0$. The similarity is then a normalized version of d_ψ. If moreover $w \circ \psi$ satisfies the condition (b) of Proposition 1, then this normalized version is a pseudo-metric.

We also have the following additional properties.

Proposition 7. *Let w and ψ defined on $(\mathcal{T}, \preceq)$ as in Definition 2, and d and d_ψ as in Theorem 1 and Proposition 6.*

- *Two elements of the lattice that are equivalent up to ψ have a zero distance: $\forall (x,y) \in \mathcal{T}^2, \psi(x) = \psi(y) \Rightarrow d_\psi(x,y) = 0$.*
- *If ψ is a morphological filter (i.e. increasing and idempotent), then $\forall (x,y) \in \mathcal{T}^2, x \preceq y \Rightarrow d_\psi(x,y) = w(\psi(y)) - w(\psi(x))$, and $d_{\psi\psi} = d_\psi$.*
- *If ψ is moreover anti-extensive (i.e. ψ is an opening), then $\forall x \in \mathcal{T}, d(x, \psi(x)) = w(x) - w(\psi(x))$. If ψ is extensive (i.e. ψ is a closing), then $\forall x \in \mathcal{T}$, $d(x, \psi(x)) = w(\psi(x)) - w(x)$.*
- *Let us denote by $Inv(\psi)$ the set of invariants by ψ (i.e. $x \in Inv(\psi) \Leftrightarrow \psi(x) = x$). We have: $\forall (x,y) \in Inv(\psi)^2, d_\psi(x,y) = d(x,y)$.*

The interest of the definitions and results of this section is that similarity and metrics are defined up to a transformation, which makes the results robust to variations of hypergraphs encoded by this transformation. The case where ψ is a filter is then of particular interest.

4.3 Example for a Dilation on $(\mathcal{P}(E), \subseteq)$

Let us first consider the simple example introduced in [3]. For any hypergraph (V, E), we define a dilation δ on $(\mathcal{P}(E), \subseteq)$ as: $\forall A \subseteq E, \delta(A) = \{e \in E \mid v(A) \cap v(e) \neq \emptyset\}$ where $v(A) = \cup_{e' \in A} v(e')$. Let $H_1 = (V, E_1)$ and $H_2 = (V, E_2)$ be two hypergraphs without empty hyperedge and δ_{E_1} and δ_{E_2} dilations defined on the set of hyperedges of H_1 and H_2, as above. We define a similarity function s by: $\forall A_1 \subseteq E_1, \forall A_2 \subseteq E_2, s(A_1, A_2) = \frac{|\delta_{E_1}(A_1) \cap \delta_{E_2}(A_2)|}{|\delta_{E_1}(A_1) \cup \delta_{E_2}(A_2)|}$, which corresponds to the similarity introduced in Definition 2 for $w = |.|$ and $\psi = \delta$.

Let us consider an example where hypergraphs are defined to represent image information. Vertices are pixels of the image, and hyperedges are subsets of pixels. Let us assume that the two images have the same support, and hence the corresponding hypergraphs have the same set of vertices. Let us denote them by $H_1 = (V, E_1)$ and $H_2 = (V, E_2)$. In this example, the hyperedge were built from color and connectivity relations as follows: we define a neighborhood of each pixel x as $\Gamma_{\alpha,\beta}(x) = \{x' \mid d_C(\mathcal{I}(x), \mathcal{I}(x')) < \alpha$ and $d_N(x, x') \leq \beta\}$, were d_C denotes a distance in the color space (or gray scale), I denotes the color of the intensity function, d_N denotes the distance in the spatial domain and α and β are two parameters to tune the extent of the neighborhood. The set of hyperedges is then defined as the set of $\Gamma_{\alpha,\beta}(x)$ for all $x \in V$. A weighted average similarity can be defined as: $s(H_1, H_2) = \frac{1}{2}(\frac{1}{\sum_{e\in E_1}|\delta_{E_1}(e)|}\sum_{e\in E_1} s(e, E_2)|\delta_{E_1}(e)| + \frac{1}{\sum_{e'\in E_2}|\delta_{E_2}(e')|}\sum_{e'\in E_2} s(e', E_1)|\delta_{E_2}(e')|)$ where $s(e, E_2) = \max_{e'\in E_2}\frac{|\delta_{E_1}(e)\cap\delta_{E_2}(e')|}{|\delta_{E_1}(e)\cup\delta_{E_2}(e')|}$ and a similar expression for $s(e', E_1)$. In the example in Figure 1(a), the similarity between the left image and its modification with an additional line is equal to 0.96. The figure on the right illustrates the dissimilarity between the two images. The dilation leads to more robustness to small and non relevant variations in the images (without the dilation, the similarity would be 0.94). Similarly, the similarity is computed between registered x-ray images of normal (b) and pathological (c) lungs, highlighting the pathological region (d). Its value is 0.75 (and 0.61 without dilation).

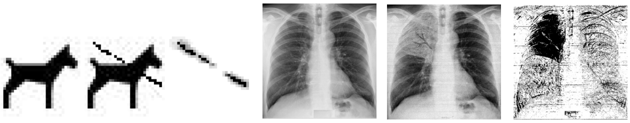

(a) (b) (c) (d)

Fig. 1. (a) An image and a modified version where a line has been introduced. The image on the right illustrates the dissimilarity (darkest grey levels). The global similarity is 0.96. Similarity (d) between normal (b) and pathological (c) lungs.

Another example is illustrated in Figure 2, where two images exhibiting some differences are compared. The comparison is illustrated in four sub-images. The similarity is equal to 1 in the top left part, to 0.75 in the top right part, to 0.98 in the bottom left part and to 0.97 in the bottom right part. Again this fits what could be intuitively expected. The global similarity, computed over the whole images, is equal to 0.93. The subdivision (even very simple here) allows us to better localize the differences.

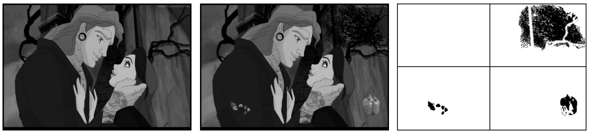


Fig. 2. Two images with some differences, and dissimilarity image

4.4 Example for an Opening on $\mathcal{T}_1 = (\mathcal{P}(\mathcal{V}), \subseteq)$

Let us now consider $\mathcal{T}_1 = (\mathcal{P}(\mathcal{V}), \subseteq)$ and $\mathcal{T}_2 = (\mathcal{P}(\mathcal{E}), \subseteq)$. As in [3], we define a dilation from $\mathcal{T}_2$ into $\mathcal{T}_1$ as: $\forall e \in E, B_e = \delta(\{e\}) = \{x \in \mathcal{V} \mid \exists e' \in \mathcal{E}, x \in v(e')$ and $v(e) \cap v(e') \neq \emptyset\} = \cup\{v(e') \mid v(e') \cap v(e) \neq \emptyset\}$, and the dilation of any subset of $\mathcal{E}$ is defined using the sup-generating property. The adjoint erosion ε, from $\mathcal{T}_1$ into $\mathcal{T}_2$ is given by: $\forall V \in \mathcal{P}(\mathcal{V}),\ \varepsilon(V) = \cup\{E \in \mathcal{P}(\mathcal{E}) \mid \forall e \in E,\ \delta(\{e\}) \subseteq V\} = \{e \in \mathcal{E} \mid \forall e' \in \mathcal{E},\ v(e') \cap v(e) \neq \emptyset \Rightarrow v(e') \subseteq V\}$.

Proposition 8. *The opening $\gamma = \delta\varepsilon$ is defined from $\mathcal{T}_1$ into $\mathcal{T}_1$ and is expressed as: $\forall V \in \mathcal{P}(\mathcal{V}), \gamma(V) = \cup\{v(e') \mid \exists e \in \varepsilon(V), v(e') \cap v(e) \neq \emptyset\} = \cup\{B_e \mid v(e) \subseteq V, B_e \subseteq V\}$.*

It is the set of vertices of the hyperedges whose neighbors (as defined by B_e) are in V and vertices of these neighbors. The example in Figure 3 illustrates that vertices that belong to "incomplete" hyperedges (i.e. for which the set of vertices is not completely included in V) are removed. This can be used for filtering hypergraphs by keeping only vertices of complete hyperedges (e_4 and e_5 here), which can be interesting for indexing and retrieval purposes (vertices from incomplete hyperedges being then considered as noise).

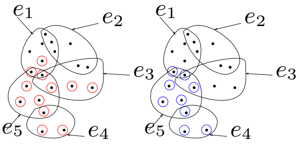


Fig. 3. Example of an opening from $\mathcal{T}_1 = (\mathcal{P}(\mathcal{V}), \subseteq)$ into $\mathcal{T}_1 = (\mathcal{P}(\mathcal{V}), \subseteq)$ [3]. The red circled vertices on the left represent V. Its opening is shown in blue on the right.

When computing the similarity $s(V, V') = \frac{|\gamma(V) \cap \gamma(V')|}{|\gamma(V) \cup \gamma(V')|}$, it is clear that if V' differs from V only by vertices from incomplete hyperedges, then $s(V, V') = 1$. The similarity is then robust to noise vertices. In particular $s(V, \gamma(V)) = 1$ since $\gamma(\gamma(V)) = V$. Other examples are shown in Figure 4, which have the same opening as in Figure 3 (right). Hence all these subsets of vertices have a similarity equal to 1 (i.e. they are equivalent up to γ and only differ by their isolated vertices).

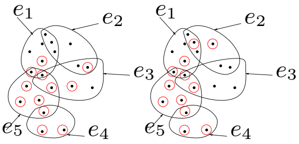


Fig. 4. Two other subsets of $\mathcal{V}$ having the same opening (shown in blue on the right in Figure 3), i.e. vertices of e_4 and e_5

Let us now consider another example of opening.

Proposition 9. *The operator γ' from $\mathcal{T}_1$ into $\mathcal{T}_1$ defined as $\forall V \in \mathcal{P}(\mathcal{V}), \gamma'(V) = \cup\{v(e) \mid v(e) \subseteq V\}$ is an opening.*

This opening keeps all vertices of complete hyperedges, i.e. the ones that are "well connected" in the hypergraph. Invariants of γ' are the subsets V that contain only vertices of complete hyperedges. An example is illustrated in Figure 5. The subset V is shown in red on the left and its opening $\gamma'(V) = v(e_5)$ in blue on the right. Note that for this example we have $\gamma(V) = \emptyset$, thus illustrating the difference between γ and γ'.

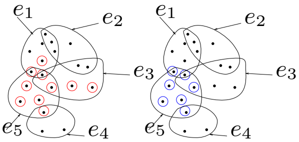


Fig. 5. Subset V (in red) and its opening $\gamma'(V)$ (in blue)

Again this makes the similarity robust to vertices which belong to incomplete hyperedges. We have $s(V, V') = 0$ iff $V \cap V'$ is the set of noise vertices.

If we consider a binary version of the similarity, i.e. V and V' are equivalent iff $\gamma'(V) = \gamma'(V')$, then equivalence classes are built of subsets of $\mathcal{V}$ which contain the vertices of the same complete hyperedges. In particular V and $\gamma'(V)$ belong to the same equivalence class. Using this equivalence relation can be useful for robust indexing and retrieval, for robust entropy definition, etc.

4.5 Example on $\mathcal{T}_3 = (\{H\}, \preceq)$

Let us now consider the most interesting case where $\mathcal{T}$ is the lattice of hypergraphs $\mathcal{T}_3 = (\{H\}, \preceq)$. We again consider opening.

Proposition 10. *The operator γ_1 defined for each hypergraph $H = (V, E)$ in $\mathcal{T}_3$ by $\gamma_1(H) = (\cup_{e \in E} v(e), E) = (V \setminus V_{\setminus E}, E)$ is an opening.*

Let us now consider the dilation introduced in [3] on this lattice. The canonical decomposition of H, from its sup generating property, is expressed as: $H = (\vee_{e \in E}(v(e), \{e\})) \vee (\vee_{x \in V_{\setminus E}}(\{x\}, \emptyset))$. From this decomposition, a dilation is defined as: $\forall x \in V_{\setminus E},\ \delta(\{x\}, \emptyset) = (\{x\}, \emptyset)$, for isolated vertices, and for elementary hypergraphs associated with hyperedges: $\forall e \in E,\ \delta(v(e), \{e\}) = (\cup\{v(e') \mid v(e') \cap v(e) \neq \emptyset\}, \{e' \in \mathcal{E} \mid v(e') \cap v(e) \neq \emptyset\})$. The dilation of any H is then derived from its decomposition and from the commutativity of dilation with the supremum.

In the particular case where H has no isolated vertices, then it is sufficient to consider the hyperedges (since the set of vertices is automatically equal to $\cup_{e \in E} v(e)$), and δ can be written in a simpler form as $\delta(\{e\}) = B_e = \{e' \in \mathcal{E} \mid v(e) \cap v(e') \neq \emptyset\}$, and $\delta(E) = \cup_{e \in E} \delta(\{e\}) = \{e' \in E \mid \exists e \in E, v(e') \cap v(e) \neq \emptyset\}$. An example is illustrated in Figure 6.

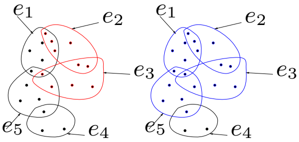


Fig. 6. The figure on the left represents $\mathcal{V}$ (vertices represented as points) and $\mathcal{E}$ (hyperedges represented as closed lines). The red lines indicate the hyperedges of H. The vertices of H are the points enclosed in these lines. The blue lines on the right represent the hyperedges of $\delta(H)$ and its vertices are the points enclosed in these lines. For this example, $\varepsilon(H)$ and $\gamma_2(H)$ are empty.

Proposition 11. *Let us consider hypergraphs without isolated vertices. The adjoint erosion of δ is given by: $\forall E \in \mathcal{P}(\mathcal{E}), \varepsilon(E) = \cup\{E' \in \mathcal{P}(\mathcal{E}) \mid \delta(E') \subseteq E\} = \{e' \in \mathcal{E} \mid B_e \subseteq E\}$ and $\varepsilon(H) = (\cup_{e \in \varepsilon(E)} v(e), \varepsilon(E))$. The opening $\gamma_2 = \delta\varepsilon$ is then $\gamma_2(E) = \cup_{B_e \subseteq E} B_e$, and $\gamma_2(H) = (\cup_{e \in \gamma_2(E)} v(e), \gamma_2(E))$.*

This result is similar to Proposition 8 on the lattices built on vertices.

In Figure 6, the erosion of H shown in red is empty, and the opening is empty as well. In Figure 7, the erosion of H is equal to $(v(e_1), \{e_1\})$ and the opening is $\gamma_2(H) = H$. Another example shows the filtering effect of this opening.

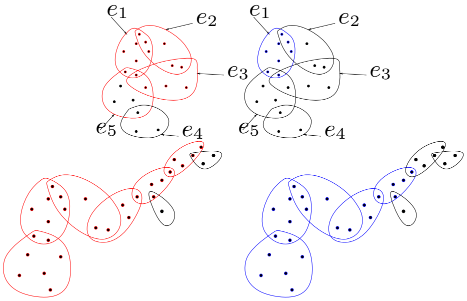


Fig. 7. Top: H is represented on the left using the same conventions as in Figure 6. Its erosion is shown on the right and $\gamma_2(H) = H$. Bottom: H and its opening $\gamma_2(H)$.

If we now consider the more general case where the hypergraphs have isolated vertices, since these are preserved by dilation, they are also preserved in the adjoint erosion and in the derived opening. These isolated vertices do not induce any change when using γ_2 or not for computing the similarity or the distance. For instance if H' is equal to H plus k additional isolated vertices, then $d(H, H') = d_{\gamma_2}(H, H') = k$. Let us now consider as a valuation on $\mathcal{T}_3$ the height. As shown in Proposition 4, we have $\forall H = (V, E) \in \mathcal{T}_3, w(H) = |V| + |E|$. We have: $d(H, \gamma_1(H)) = |V_{\setminus E}|$, which is the number of isolated vertices in H (the distance evaluates the amount of "noise" in H if isolated vertices are interpreted as noise vertices). If H and H' differ only by isolated vertices, then $d_{\gamma_1}(H, H') = 0$. If we consider now γ_2, then the general results expressed in Proposition 7 hold, along with the associated interpretation. Let us give a few simple examples: For the first example of H depicted in Figure 7, $d(H, \gamma_2(H)) = d(H, H) = 0$. For the second example of H, $d(H, \gamma_2(H)) = 4 + 1 = 5$. Two hypergraphs H_1 and H_2 having the same opening by γ_2 are displayed in Figure 8. Hence $d_{\gamma_2}(H_1, H_2) = 0$. Now if k isolated vertices are added to one of the two hypergraphs, their opening will stay the same up to these isolated vertices, and $d_{\gamma_2}(H_1, H_2) = k$.

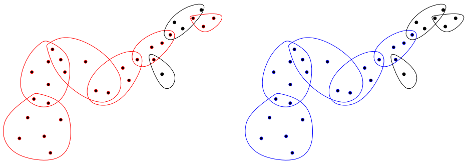


Fig. 8. Two hypergraphs H_1 (left) and H_2 (right). Their openings are $\gamma_2(H_1) = \gamma_2(H_2) = H_2$ and $d_{\gamma_2}(H_1, H_2) = 0$.

5 Conclusion

The proposed framework offers new tools for defining similarity measures and pseudo-metrics, which are robust to variations (encoded by morphological operators) of hypergraphs. They can be incorporated in existing systems for hypergraph-based feature selection, indexing, retrieval, matching. As an example, let us consider the equivalence relation on any lattice of hypergraphs $\mathcal{T}$ defined by $\forall (x, y) \in \mathcal{T}^2, xRy \Leftrightarrow \psi(x) = \psi(y)$ where ψ is a morphological operator on $\mathcal{T}$. This equivalence relation induces a partition of $\mathcal{T}$, denoted by $\mathcal{T} = \cup_i T_i$. A discrete probability distribution can then be defined as $p_i = \frac{|T_i|}{|\mathcal{T}|}$ from which an entropy of $\mathcal{T}$ (up to ψ) can be derived. This defines a new entropic criterion that can be used in feature selection methods such as [24]. Future work aims at exploring other examples of morphological operators in the proposed framework (for instance as the ones defined in [9] on simplicial complexes), and weaker forms of valuations, by considering the sub- or supra-modular cases [20].

References

1. Birkhoff, G.: Lattice theory, 3rd edn., vol. 25. American Mathematical Society (1979)
2. Bloch, I., Bretto, A.: Mathematical Morphology on Hypergraphs: Preliminary Definitions and Results. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) DGCI 2011. LNCS, vol. 6607, pp. 429–440. Springer, Heidelberg (2011)
3. Bloch, I., Bretto, A.: Mathematical morphology on hypergraphs, application to similarity and positive kernel. Computer Vision and Image Understanding 117(4), 342–354 (2013)
4. Bloch, I., Heijmans, H., Ronse, C.: Mathematical Morphology. In: Aiello, M., Pratt-Hartman, I., van Benthem, J. (eds.) Handbook of Spatial Logics, ch. 13, pp. 857–947. Springer (2007)

5. Bretto, A.: Hypergraph Theory: an Introduction. Mathematical Engineering. Springer (2013)
6. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based classification: Concepts and algorithms. Journal of Machine Learning Research 10, 747–776 (2009)
7. Choi, S.S., Cha, S.H., Tappert, C.C.: A survey of binary similarity and distance measures. Journal of Systemics, Cybernetics and Informatics 8(1), 43–48 (2010)
8. Cousty, J., Najman, L., Dias, F., Serra, J.: Morphological filtering on graphs. Computer Vision and Image Understanding 117, 370–385 (2013)
9. Dias, F., Cousty, J., Najman, L.: Some Morphological Operators on Simplicial Complex Spaces. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) DGCI 2011. LNCS, vol. 6607, pp. 441–452. Springer, Heidelberg (2011)
10. Duchenne, O., Bach, F., Kweon, I., Ponce, J.: A tensor-based algorithm for high-order graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(12), 2383–2395 (2011)
11. Gao, Y., Wang, M., Tao, D., Ji, R., Dai, Q.: 3-D object retrieval and recognition with hypergraph analysis. IEEE Transactions on Image Processing 21(9), 4290–4303 (2012)
12. Heijmans, H.J.A.M.: Morphological Image Operators. Academic Press, Boston (1994)
13. 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(3), 245–295 (1990)
14. Jouili, S., Tabbone, S.: Hypergraph-based image retrieval for graph-based representation. Pattern Recognition 45, 4054–4068 (2012)
15. Liang, Z., Chi, Z., Fu, H., Feng, D.: Salient object detection using content-sensitive hypergraph representation and partitioning. Pattern Recognition 45, 3886–3901 (2012)
16. Loménie, N., Stamon, G.: Morphological mesh filtering and α-objects. Pattern Recognition Letters 29(10), 1571–1579 (2008)
17. Meyer, F., Stawiaski, J.: Morphology on Graphs and Minimum Spanning Trees. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 161–170. Springer, Heidelberg (2009)
18. Ronse, C., Heijmans, H.J.A.M.: The Algebraic Basis of Mathematical Morphology – Part II: Openings and Closings. Computer Vision, Graphics and Image Processing 54(1), 74–97 (1991)
19. Serra, J. (ed.): Image Analysis and Mathematical Morphology, Part II: Theoretical Advances. Academic Press, London (1988)
20. Simovici, D.: Betweenness, metrics and entropies in lattices. In: 38th IEEE International Symposium on Multiple Valued Logic, ISMVL 2008. pp. 26–31 (2008)
21. Ta, V.-T., Elmoataz, A., Lézoray, O.: Partial Difference Equations over Graphs: Morphological Processing of Arbitrary Discrete Data. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 668–680. Springer, Heidelberg (2008)
22. Vincent, L.: Graphs and Mathematical Morphology. Signal Processing 16(4), 365–388 (1989)
23. Voloshin, V.I.: Introduction to Graph and Hypergraph Theory. Nova Science Publishers (2009)
24. Zhang, Z., Hancock, E.: Hypergraph based information-theoretic feature selection. Pattern Recognition Letters 33, 1991–1999 (2012)

Simplification Operators on a Dimension-Independent Graph-Based Representation of Morse Complexes

Lidija Čomić[1], Leila De Floriani[2], and Federico Iuricich[2]

[1] University of Novi Sad - Faculty of Technical Sciences
comic@uns.ac.rs
[2] University of Genova - Department of Computer Science
deflo@disi.unige.it
federico.iuricich@unige.it

Abstract. Ascending and descending Morse complexes are defined by the critical points and integral lines of a scalar field f defined on a manifold M. They induce a subdivision of M into regions of uniform gradient flow, thus providing a compact description of the topology of M and of the behavior of f over M. We represent the ascending and descending Morse complexes of f as a graph, that we call the *Morse incidence graph (MIG)*. We have defined a simplification operator on the graph-based representation, which is atomic and dimension-independent, and we compare this operator with a previous approach to the simplification of 3D Morse complexes based on the cancellation operator. We have developed a simplification algorithm based on a simplification operator, which operates on the MIG, and we show results from this implementation as well as comparisons with the cancellation operator in 3D.

Keywords: geometric modeling, Morse theory, Morse complexes, simplification.

1 Introduction

Representing topological information extracted from discrete scalar fields is a relevant issue in several applications, such as terrain modeling, or analysis and visualization of static and time-varying 3D scalar fields arising from physical simulation or describing medical data. Morse theory offers a natural and intuitive way of analyzing the structure of a scalar field f as well as of compactly representing it through decompositions of its domain M into meaningful regions associated with the critical points of f, giving rise to the *Morse* and *Morse–Smale complexes*. *Descending* and *ascending* Morse complexes decompose M into cells defined by the integral lines of f converging to and originating at critical points of f, respectively. The Morse–Smale complex decomposes M into cells defined by integral lines with the same origin and destination. We represent here the topology of the descending and ascending Morse complexes in arbitrary dimensions as a graph, that we call the *Morse Incidence Graph* (*MIG*), in which the

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 13–24, 2013.

nodes encode the cells of the Morse complexes, and the arcs encode their mutual incidence relations. The MIG provides also a combinatorial description of the Morse–Smale complex.

The main issue in the usage of Morse and Morse–Smale complexes in real-world applications scenarios lies in their storage requirements and computation costs when extracted from very large data sets describing 2D and 3D scalar fields, which are common in current applications. Thus, simplification of such complexes by reducing the number of cells and their mutual relations becomes a must. A separate, but still related, issue is the presence of noise in the data, which leads to over-segmentation. In this case as well, we need operators for simplifying Morse and Morse–Smale complexes and their combinatorial representation.

There have been two approaches in the literature to the simplification of Morse and Morse–Smale complexes. The approach in [13] is specific for 3D Morse–Smale complexes and is based on the cancellation operator defined in Morse theory [15]. The major problem with using cancellation is that it may increase the size of the Morse–Smale complexes when the cancellation does not involve a minimum or a maximum, thus causing memory problems when dealing with large-size data sets [11]. The approach in [4,5] is based on an atomic simplification operator, called *remove*, which is entirely dimension-independent, never increases the size of the complexes, and defines a minimally complete basis for expressing any simplification operator on such complexes.

Here, we report the effect of *remove* operator on the MIG. We show that its effect on the MIG is always local, and this is true in any dimension. Moreover, it is simple to implement in a completely dimension-independent way. We have also implemented the cancellation operator on the MIG in the 3D case and compared it with the 3D instances of the *remove* operator. We show that the size of the simplified MIG produced by *remove* is always smaller than that of the graph produced by cancellation.

The remainder of this paper is organized as follows. In Section 2, we review some basic notions on Morse theory and Morse complexes, some algorithms for their computation, and the relation between Morse complexes and the watershed decomposition. In Section 3, we describe a dual incidence-based graph representation of the Morse complexes, the *Morse Incidence Graph* (MIG). In Section 4, we recall the definition of the *remove* operator on the scalar field and we describe its effect on the MIG. In Section 5, we describe the effect of the cancellation operator in [13] on the MIG. In Section 6, we present experimental results on the behavior of the simplification algorithm based on *remove* and on the cancellation operators, and comparisons between our simplification operator and cancellation in 3D. Finally, in Section 7, we make some concluding remarks.

2 Background Notions and Related Work

Morse theory [15] captures the relation between the topology of a manifold M and the critical points of a scalar (real-valued) function f defined on M.

Let f be a C^2 real-valued function defined over a closed compact n-manifold M. A point p is a *critical point* of f if and only if the gradient $\nabla f = (\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n})$

(in some local coordinate system around p) of f vanishes at p. A function f is a *Morse function* if and only if all its critical points are non-degenerate (i.e., the Hessian matrix $Hess_p f$ of the second derivatives of f at a point p is non-singular). The number i of negative eigenvalues of $Hess_p f$ at p is called the *index* of p, and p is called an *i-saddle.* A 0-saddle is also called a *(local) minimum*, and an n-saddle a *(local) maximum.* An *integral line* of f is a maximal path which is everywhere tangent to ∇f. Each integral line starts and ends at critical points of f, called its *origin* and its *destination.*

Integral lines that converge to a critical point p of index i form an i-cell called a *descending cell*, or *manifold*, of p. Dually, integral lines that originate at p form its ascending $(n-i)$-cell. The descending and ascending cells decompose M into *descending* and *ascending Morse complexes*, denoted as Γ_d and Γ_a, respectively (see Figure 1 (a) and (b)). A Morse function f is called a *Morse–Smale function* if and only if each non-empty intersection of a descending and an ascending manifold is transversal. The connected components of the intersection define a *Morse–Smale complex* (see Figure 1 (c)).

The decomposition into ascending Morse complexes in 2D is related to the watershed decomposition, developed for image analysis. If f is a function which has a gradient ∇f everywhere except possibly at some isolated points, then the *topographic distance* $T_D(p,q)$ between two points p,q belonging to the domain D of f is defined as $T_D(p,q) = \inf_P \int_P |\nabla f(P(s))| ds$ [16,18]. If there is an integral line which reaches both p and q, then the topographic distance between these two points is equal to the difference in elevation between them. Otherwise, it is strictly greater than the difference in elevation. The *catchment basin* $CB(m_i)$ of a minimum m_i is the set of points which are closer (in the sense of topographic distance) to m_i than to any other minimum. The *watershed* $WS(f)$ of f is the set of points which do not belong to any catchment basin. When f is a Morse function, then the catchment basins of the minima of f correspond to the 2-cells in the ascending Morse complex of f.

Topological watershed [1] is obtained by lowering (until idempotence) some points of the original image while preserving the connectivity of each lower cross-section. This operator does not preserve the connected components or holes in a 3D object. The catchment basins and the watershed separating them are composed of pixels (2-cells) in the image, while in the ascending Morse complexes ascending 2-cells (corresponding to minima) are separated by 1-cells (corresponding to saddles).

Algorithms for decomposing the domain M of f into an approximation of a Morse, or of a Morse–Smale complex in 2D can be classified as *boundary-based* or *region-based.* In [7,6,14], algorithms for extracting the Morse–Smale complex from a tetrahedral mesh have been proposed. Discrete methods rooted in the discrete Morse theory proposed by Forman [9] are computationally more efficient [10,17]. For a survey, see [2,3]. Alternative region-based techniques for computing the ascending and descending Morse complexes are based on the discrete watershed transform (see [18] for a survey).

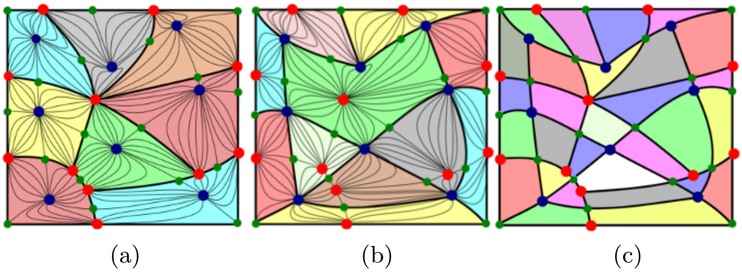


Fig. 1. A portion of an ascending Morse complex in 2D (a); the dual descending Morse complex (b); the corresponding Morse–Smale complex (c).

3 The Morse Incidence Graph (*MIG*)

The topology of complexes Γ_a and Γ_d is represented in the form of a graph, called the *Morse Incidence Graph (MIG)* $G = (N, A, \psi)$, where N is the set of nodes, A is the set of arcs, and $\psi : A \to \mathbb{N}$ is a labeling function, such that:

1. the set N of nodes is partitioned into $n+1$ subsets N_0, N_1,...,N_n, such that there is a one-to-one correspondence between the nodes in N_i (which we call *i-nodes*) and the i-cells of Γ_d, (and thus the $(n-i)$-cells of Γ_a);
2. there is an arc (p, q) joining an i-node p with an $(i+1)$-node q if and only if i-cell p is on the boundary of $(i+1)$-cell q in Γ_d, (and thus q is on the boundary of p in Γ_a);
3. each arc (p, q) is labeled with the number $\psi(p, q)$ of times i-cell p (corresponding to i-node p) in Γ_d is incident into $(i+1)$-cell q (corresponding to $(i+1)$-node q) in Γ_d, and thus $\psi(p, q)$ is equal to the number of integral lines of f connecting i-saddle p to $(i+1)$-saddle q.

Each node is labeled with the critical point (or equivalently, the descending cell) it represents. If f is a Morse–Smale function, then the *MIG* provides also a combinatorial representation of the 1-skeleton of its Morse–Smale complex. Figure 2 illustrates the descending and ascending complexes, and the corresponding *MIG* for function $f(x, y) = \cos x \cos y$.

4 The *Remove* Simplification Operator

In [4,5], we have introduced a dimension-independent simplification operator called *remove* on the Morse complexes and on the *MIG* representing them. This operator (together with its inverse one) forms a basis for defining any operator that updates Morse and Morse–Smale complexes on a manifold M in a topologically consistent manner. Operator *remove* has two instances, namely $remove_{i,i+1}$ and $remove_{i,i-1}$, for $1 \le i \le n-1$. We describe the effect of the *remove* operator on the Morse function f and on the corresponding *MIG*.

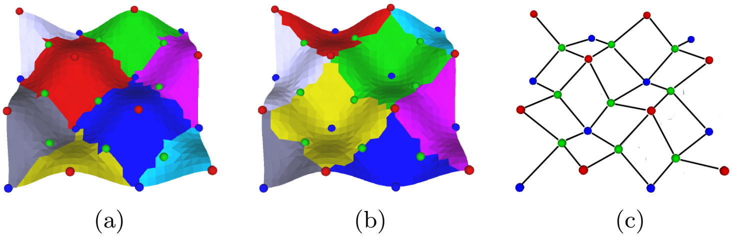


Fig. 2. The descending (a) and ascending (b) 2D Morse complex for function $f(x, y) = \cos x \cos y$ and the corresponding *MIG* (c).

4.1 *Remove* on the Scalar Field

A $remove_{i,i+1}$ operator collapses an i-saddle q and an $(i+1)$-saddle p, that are connected through a unique integral line. It is defined if i-saddle q is connected to at most one other $(i+1)$-saddle p' different from $(i+1)$-saddle p. There are two types of $remove_{i,i+1}$, denoted as $remove_{i,i+1}(q, p, p')$ and $remove_{i,i+1}(q, p, \emptyset)$, respectively. $Remove_{i,i+1}(q, p, p')$ applies when the i-saddle q is connected to the $(i+1)$-saddle p and exactly one other $(i+1)$-saddle p' different from p. It collapses the i-saddle q and the $(i+1)$-saddle p into the $(i+1)$-saddle p'. $Remove_{i,i+1}(q, p, \emptyset)$ deals with the situation in which the i-saddle q is connected to only one $(i+1)$-saddle p. It eliminates the i-saddle q and the $(i+1)$-saddle p from the set of critical points of the scalar field f.

$Remove_{i,i-1}(q, p, p')$ operator is dual to the previous one. It collapses an i-saddle q and an $(i-1)$-saddle p that are connected through a unique integral line. As for $remove_{i,i+1}$, there are two types of $remove_{i,i-1}$ operator, denoted as $remove_{i,i-1}(q, p, p')$ and $remove_{i,i-1}(q, p, \emptyset)$, respectively. $Remove_{i,i-1}(q, p, p')$ applies when the i-saddle q is connected to the $(i-1)$-saddle p and exactly one other $(i-1)$-saddle p' different from p. It collapses the i-saddle q and $(i-1)$-saddle p into the $(i-1)$-saddle p'. $Remove_{i,i-1}(q, p, \emptyset)$ eliminates the i-saddle q and the unique $(i-1)$-saddle p connected to q.

In 2D, we have two *remove* operators, both deleting a saddle and an extremum. $Remove_{1,2}(q, p, p')$ consists of collapsing a maximum (2-saddle) p and a saddle (1-saddle) q into a maximum (2-saddle) p', and $remove_{1,0}(q, p, p')$ consists of collapsing a minimum (0-saddle) p and a saddle (1-saddle) q into a minimum (0-saddle) p'. $Remove_{1,2}(q, p, \emptyset)$ consists of deleting a maximum (2-saddle) p and a saddle (1-saddle) q. Dually, $remove_{1,0}(q, p, \emptyset)$, consists of deleting a minimum (0-saddle) p and a saddle (1-saddle) q.

In 3D there are four *remove* operators, two $remove_{i,i+1}$ and two $remove_{i,i-1}$ operators. $Remove_{2,3}(q, p, p')$ collapses a maximum (3-saddle) p and a 2-saddle q into a maximum (3-saddle) p', while $remove_{1,2}(q, p, p')$, collapses a 2-saddle p and a 1-saddle q into a 2-saddle p'. Dually, $remove_{1,0}(q, p, p')$ collapses a minimum (0-saddle) p and a 1-saddle q into a minimum (0-saddle) p', while $remove_{2,1}(q, p, p')$ collapses a 1-saddle p and a 2-saddle q into a 1-saddle p'. The operators of the second type delete the two critical points q and p.

4.2 *Remove* on the *MIG*

On a MIG $G = (N, A, \psi)$, the effect of the *remove* operator is to transform G into a simplified MIG $G' = (N', A', \psi')$ by eliminating two nodes p and q from N, and reconnecting the remaining nodes in the simplified graph G' so that $|N'| = |N| - 2$ and $|A'| < |A|$. Let us consider the effect of $remove_{i,i+1}(q, p, p')$. We consider the following sets of nodes in the MIG $G = (N, A, \psi)$:

- $Z = \{z_h, h = 1, .., h_{max}\}$ is the set of the $(i-1)$-nodes connected to the i-node q;
- $S = \{s_k, k = 1, .., k_{max}\}$ is the set of the $(i+2)$-nodes connected to the $(i+1)$-node p;
- $R = \{r_j, j = 1, .., j_{max}\}$ is the set of the i-nodes connected to the $(i+1)$-node p (the set R may be empty).

The set S is empty for $i = n - 1$. For $remove_{i,i+1}(q, p, \emptyset)$ operator, the sets Z, R and S are exactly the same as for $remove_{i,i+1}(q, p, p')$.

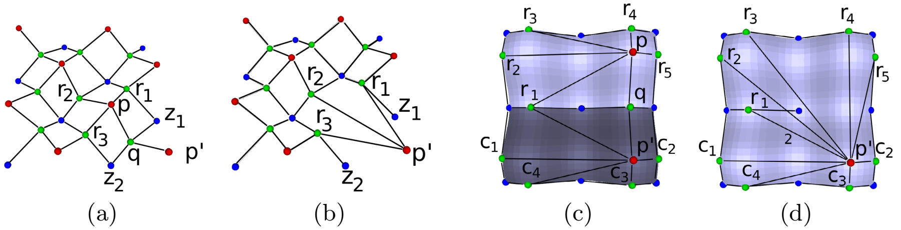


Fig. 3. An example of a $remove_{1,2}(q, p, p')$ on a MIG in the 2D case (a) and (b). MIG overlayed on the descending Morse complex before (c) and after (d) $remove_{1,2}(q, p, p')$ in 2D. The label of the arc connecting nodes r_1 and p' is increased to 2, and 1-cell r_1 appears two times on the boundary of 2-cell p' after $remove_{1,2}(q, p, p')$.

Figure 3 (a) illustrates $remove_{1,2}(q, p, p')$ in 2D, which is a maximum-saddle operator. The set Z consists of nodes z_1 and z_2, which correspond to minima. The set S is empty since the operator involves an extremum. The set R consists of saddles r_1, r_2 and r_3 connected to the maximum p.

$Remove_{i,i+1}(q, p, p')$ operator is feasible on a MIG $G = (N, A, \psi)$ if

- i-node q is connected to exactly two different $(i+1)$-nodes p and p', and
- the label of arc (p, q) is 1 $(\psi(p, q) = 1)$.

The effect of $remove_{i,i+1}(q, p, p')$ on G is to

- delete nodes p and q,
- delete all the arcs incident in either i-node q or $(i+1)$-node p,
- introduce an arc (p, r_j) for each $r_j \in R$ (if such arc does not already exist),
- set $\psi'(p', r_j) = \psi(p', q) \cdot \psi(p, r_j) + \psi(p', r_j)$.

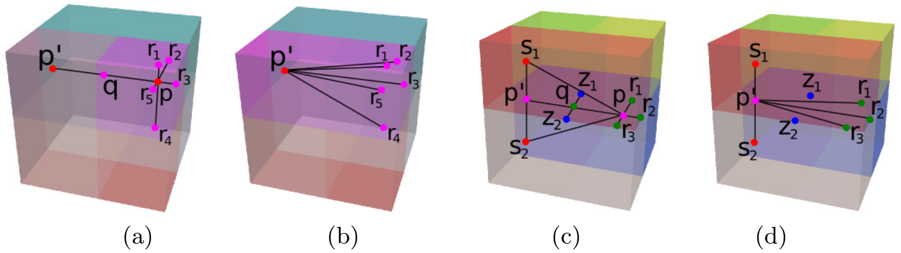


Fig. 4. The *MIG*, overlayed on the descending Morse complex, before (a) and after (b) $remove_{2,3}(q,p,p')$, and before (a) and after (b) $remove_{1,2}(q,p,p')$ in 3D.

The labels of other arcs in the simplified *MIG* $G' = (N', A', \psi')$ remain unchanged.

In the example illustrated in Figure 3 (a) and (b), $remove_{1,2}(q,p,p')$, deletes nodes p and q, as well as the arcs incident in 1-node q (arcs (q,p), (q,p'), (q,z_1) and (q,z_2)). Arcs (p,r_1), (p,r_2) and (p,r_3) are replaced with arcs (p',r_1), (p',r_2) and (p',r_3).

Figure 3 (c) and (d) illustrates the effect of $remove_{1,2}(q,p,p')$ in 2D, which is a saddle-maximum removal. Before $remove_{1,2}(q,p,p')$, 1-node r_1 is connected to 2-nodes p and p', and the labels of arcs (p,r_1) and (p',r_1) are equal to 1 ($\psi(p,r_1) = \psi(p',r_1) = 1$). After $remove_{1,2}(q,p,p')$, the label of arc (p',r_1) is equal to 2 ($\psi'(p',r_1) = \psi(p',q) \cdot \psi(p,r_1) + \psi(p',r_1) = 1 \cdot 1 + 1 = 2$). The labels of other arcs in the graph are equal to 1.

Figure 4 (a) and (b) illustrates the effect of $remove_{2,3}(q,p,p')$ in 3D, which performs the removal of a 2-saddle and a maximum. Nodes p and q are deleted as well as all the arcs incident in 2-node q. Arcs (p,r_1), (p,r_2), (p,r_3), (p,r_4) and (p,r_5) are replaced with arcs (p',r_1), (p',r_2), (p',r_3), (p',r_4) and (p',r_5).

Figure 4 (c) and (d) shows the effect of $remove_{1,2}(q,p,p')$ in 3D, which is a 1-saddle-2-saddle removal. Nodes p and q are deleted, as well as all the arcs incident in 1-node q, and arcs connecting 2-node p to 3-nodes s_1 and s_2. Arcs (p,r_1), (p,r_2) and (p,r_3) are replaced with arcs (p',r_1), (p',r_2) and (p',r_3). The dual $remove_{i,i-1}(q,p,p')$ operator can be expressed as a modification of the graph $G = (N, A, \psi)$ in a completely dual fashion.

5 Operator *i-Cancellation*

The cancellation operator [13], that we call *i-cancellation*, is a simplification operator defined in Morse theory [15]. It eliminates i-saddle q and $(i+1)$-saddle p connected through a unique integral line, but it does not impose any constraints on the number of $(i+1)$-saddles connected to q, or on the number of i-saddles connected to p. We define here the i-*cancellation*(q,p) operator on the *MIG*. We denote as $T = \{t_l, l = 1, ..., l_{max}\}$ the set of $(i+1)$-nodes different from $(i+1)$-node p and connected to i-node q. $R = \{r_j, j = 1, .., j_{max}\}$ is the set of i-nodes (i-saddles) connected to p and different from q.

An *i-cancellation*(q, p), $0 \leq i \leq n-1$, is feasible on a *MIG* $G = (N, A, \psi)$ if

- i-node q is connected to $(i+1)$-node p, and
- the label of arc (p, q) is 1 ($\psi(p, q) = 1$).

The effect of *i-cancellation*(q, p) is to

- delete i-node q and $(i+1)$-node p,
- delete all the arcs incident in either i-node q or $(i+1)$-node p,
- introduce an arc (t_l, r_j) for each $t_l \in T$ and each $r_j \in R$ (if such arc does not already exist),
- set $\psi'(t_l, r_j) = \psi(t_l, q) \cdot \psi(p, r_j) + \psi(t_l, r_j)$.

The labels of the other arcs in the simplified *MIG* $G' = (N', A', \psi')$ are the same as the labels of those arcs in G.

In the 2D case, $remove_{1,2}(q, p, p')$ is the same as the maximum-saddle 1-*cancellation*(q, p), and $remove_{1,0}(q, p, p')$ is the same as the minimum-saddle 0-*cancellation*(p, q). In 3D, $remove_{2,3}(q, p, p')$ operator involving an extremum (maximum-2-saddle) is exactly the same as the 2-*cancellation*(q, p). Dually, $remove_{1,0}(q, p, p')$ operator involving an extremum (minimum-1-saddle) is exactly the same as the 0-*cancellation*(p, q). In general, $remove_{n-1,n}(q, p, p')$ and $remove_{1,0}(q, p, p')$ operators involving an extremum p are the same as the $(n-1)$-*cancellation*(q, p) and 0-*cancellation*(p, q), respectively.

The *i-cancellation* involving only saddles (which are not extrema) is more complex. As an example, let us consider the 1-*cancellation* of a 1-saddle and a 2-saddle in 3D. This operator has been implemented in [12,11] on the 1-skeleton of the Morse–Smale complex (which is combinatorially equivalent to the *MIG* of the corresponding Morse complexes). The 1-*cancellation*(q, p) of 1-node q and 2-node p is feasible on the *MIG* $G = (N, A, \psi)$ if nodes q and p are connected, and the label of arc (p, q) is 1 ($\psi(p, q) = 1$). Let $G' = (N', A', \psi')$ be the graph after 1-*cancellation*(q, p). The effect of 1-*cancellation*(q, p) consists of deleting nodes p and q, as well as all the arcs incident in nodes p and q, and adding one arc for each pair (r_j, t_l) where r_j belongs to R and t_l belongs to T. Thus, the 1-*cancellation* operator deletes two nodes from N, but it increases the number of arcs connecting 1-nodes to 2-nodes in the graph by deleting $|R| + |T| + 1$ of such arcs, but adding $|R| * |T|$ of them. Thus, it is not a simplification operator, since it does not reduce the size of the graph. In [11], this issue has been discussed at length, since it can cause computational problems and, more importantly, make the application of *i-cancellation* operator unfeasible on large-scale data sets. Several strategies are proposed in [11], which aim at postponing an *i-cancellation* that would introduce a number of arcs greater than a predefined threshold, or vertices with valence greater than a predefined threshold. On the contrary, the *remove* operator always reduces the size of the graph.

6 Experimental Results

We have performed experiments on the simplification of Morse complexes by using six data sets describing 2D scalar fields, and eight data sets describing 3D

Table 1. Comparison of *cancellation* and *remove* operators

Name	N Simpl	Nodes	Arcs	Cost (MB)	Time (sec.)
Neghip		3K	11512	0.17	-
i-cancellation	1200	700	2621	0.04	0.68
remove	1200	700	2395	0.03	0.62
Hydrogen	-	23K	65961	1.0	-
i-cancellation	7000	9K	35123	0.53	25.1
remove	7000	9K	23091	0.35	17.86
Bucky	-	46K	157984	2.4	-
i-cancellation	7000	32K	128231	1.95	73.5
remove	7000	32K	84487	1.23	33.4
Aneurism	-	125K	1015724	15.49	-
i-cancellation	10000	105K	748192	11.41	233.28
remove	10000	105K	435910	6.65	70.54
VisMale	-	900K	3588570	54.75	-
i-cancellation	10000	880K	3513889	53.61	37.12
remove	10000	880K	3107124	47.41	9.43
Foot	-	1550K	7178384	109.5	-
i-cancellation	45000	1460K	6137199	93.64	2882.1
remove	45000	1460K	5413683	82.6	1187.3

scalar fields on a 3.2GHz processor with 2.0Gb memory. The MIG representing the Morse complexes is extracted using the algorithm described in [17].

We have developed a simplification algorithm for Morse complexes in arbitrary dimensions based on the $remove_{i,i+1}$ and $remove_{i,i-1}$. A *persistence* value is associated with any *remove* operator by considering the function values of the two critical points p and q deleted by the operator. Intuitively, the *persistence* of a pair of critical points measures the importance of the pair and is equal to the absolute difference in function values between the two points [8]. The objective of the simplification algorithm is to reduce the size of the Morse complex by removing critical points which are due to the presence of noise or which are not relevant for the need of a specific application. Simplification is also applied when the size of the original Morse complex is too large for the computation resources available.

The simplification algorithm starts by computing all feasible simplifications, evaluates their persistence and inserts them in an ordered queue in increasing order of persistence. At each step, a simplification is removed from the queue and applied to the current MIG. The process terminates when either a certain number of simplifications has been performed or when a specified value of persistence is reached.

We have also implemented a 3D version of the simplification algorithm based on cancellation. In this case, the number of critical nodes is reduced at each step, but not necessarily the number of arcs.

In the experiments, we have used different thresholds on persistence value: 1% of the max persistence value for light noise removal, 10% for stronger noise removal, and 20% or greater for consistently reducing the complexity of the

MIG. The storage cost of the simplified MIG using these three different thresholds is equal to 95%, 65% and 35% of the cost of the MIG at full resolution.

We have also analyzed the statistics on the operators involved in the simplification process for some 3D data sets. We have noticed that saddle-saddle simplification operators are likely to be performed early in the simplification process. If the simplification algorithm is based on persistence, this means that a large number of arcs will be introduced in the MIG early in the simplification process, influencing both the efficiency (speed) of the algorithm and its versatility (the number of feasible simplifications). This result underlines the importance of having an efficient operator for simplifying saddles.

In Table 1, we show the results obtained by comparing the *remove* operator with the *cancellation* operator. For each 3D data set, we show in the first row the number of nodes and arcs in the full resolution MIG. In the second and third rows, we show the statistics related to *cancellation* and *remove* operators, respectively: the number of simplifications applied, the number of nodes and arcs in the simplified MIG, the cost of the data structure encoding the MIG (in MB), and the time (in sec) needed to perform the simplifications.

The number of arcs in the graph simplified with *cancellation* always exceeds the number of arcs in the graph simplified with the same number of *remove*. Such behavior influences the efficiency of the whole algorithm, doubling the time needed to manage and enqueue a larger number of arcs (and thus, a greater number of possible simplifications) for large data sets.

When the data set is small and the number of simplifications is high compared to the total number of nodes the two methods are quite similar ($Neghip$). With the growth of the dimension of the data set the two methods start to differ ($Hydrogen$): by using *remove* we can get a 20% more compressed MIG in about half the time than by using *cancellation*. In particular, the *remove* operator is particularly useful in the first simplifications performed on a data set (simplifications that can be interpreted as noise removal). On many data sets we have noticed that by using *cancellation* the number of arcs remains approximately the same while by using *remove* their number immediately decreases ($VisMale$). In general, the cost of the MIG is reduced by 10% to 20% by using *remove* instead of *cancellation* and the same number of simplifications can be performed in half the time.

In Figure 5, we illustrate the result of our simplification algorithm on a 3D *Buckyball* data set (Figure 5 (a)), which represents the electron density inside a C-60 Buckminsterfullerene. The full-resolution MIG is shown in Figure 5 (b). The MIGs after 11K and 12K simplifications are shown in Figure 5 (c) and (d), respectively.

7 Concluding Remarks

We have described two simplification operators on the MIG representing the topology of the Morse complexes of a scalar field f. The *remove* operator always reduces the size of the MIG, both in terms of the number of nodes and of the

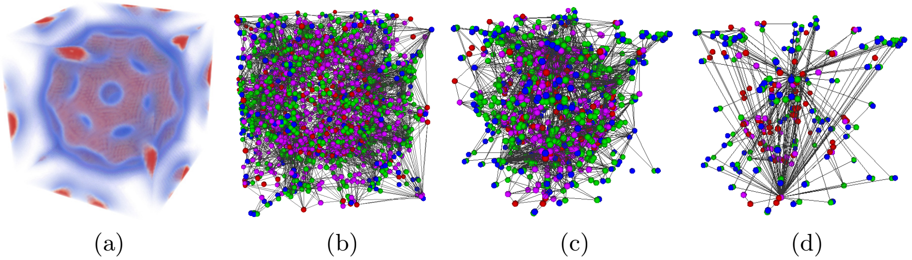


Fig. 5. The field behavior for the *Buckyball* data set (a). The *MIG* at full resolution (b), the *MIG* after 11K (c) and after 12K simplifications (d).

number of arcs in the *MIG*. The *i-cancellation* operator is guaranteed to reduce only the number of nodes in the *MIG*, but in the general case it increases the number of its arcs, thus increasing the total size of the *MIG*.

We have designed and implemented a simplification algorithm based on the two simplification operators on the *MIG*, and we have performed experiments to compare the two operators. We have shown that the number of arcs in the simplified *MIG* obtained through the *i-cancellation* operator always exceeds the number of arcs in the simplified *MIG* obtained through the *remove* operator. The large number of arcs in the *MIG* influences not only the storage cost of the data structure for encoding it and the time required for performing the simplifications, but it also reduces the number of feasible simplification and thus the flexibility of the simplification algorithm.

Based on the *remove* operator, we have designed and implemented a multi-resolution model for the Morse complexes represented as a *MIG* that we call the *Multi-Resolution Morse Incidence Graph (MMIG)* [5]. It encodes a large number of topological representations of the Morse complexes at both uniform and variable level of detail.

Acknowledgments. This work has been partially supported by the Italian Ministry of Education and Research under the PRIN 2009 program, and by the National Science Foundation under grant number IIS-1116747.

References

1. Bertrand, G.: On Topological Watersheds. Journal of Mathematical Imaging and Vision 22(2-3), 217–230 (2005)
2. Biasotti, S., De Floriani, L., Falcidieno, B., Papaleo, L.: Morphological Representations of Scalar Fields. In: De Floriani, L., Spagnuolo, M. (eds.) Shape Analysis and Structuring, pp. 185–213. Springer (2008)
3. Biasotti, S., Floriani, L.D., Falcidieno, B., Frosini, P., Giorgi, D., Landi, C., Papaleo, L., Spagnuolo, M.: Describing shapes by geometrical-topological properties of real functions. ACM Comput. Surv. 40, Article 12 (2008)

4. Čomić, L., De Floriani, L., Iuricich, F.: Simplifying Morphological Representations of 2D and 3D Scalar Fields. In: 19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2011, pp. 437–440. ACM (2011)
5. Čomić, L., De Floriani, L., Iuricich, F.: Dimension-Independent Multi-Resolution Morse Complexes. Computers & Graphics 36(5), 541–547 (2012)
6. Edelsbrunner, H., Harer, J.: The Persistent Morse Complex Segmentation of a 3-Manifold. In: Magnenat-Thalmann, N. (ed.) 3DPH 2009. LNCS, vol. 5903, pp. 36–50. Springer, Heidelberg (2009)
7. Edelsbrunner, H., Harer, J., Natarajan, V., Pascucci, V.: Morse-Smale Complexes for Piecewise Linear 3-Manifolds. In: Proceedings 19th ACM Symposium on Computational Geometry, pp. 361–370 (2003)
8. Edelsbrunner, H., Harer, J., Zomorodian, A.: Hierarchical Morse Complexes for Piecewise Linear 2-Manifolds. In: Proceedings 17th ACM Symposium on Computational Geometry, pp. 70–79 (2001)
9. Forman, R.: Morse Theory for Cell Complexes. Advances in Mathematics 134, 90–145 (1998)
10. Gyulassy, A., Bremer, P.-T., Hamann, B., Pascucci, V.: A Practical Approach to Morse-Smale Complex Computation: Scalability and Generality. IEEE Transactions on Visualization and Computer Graphics 14(6), 1619–1626 (2008)
11. Gyulassy, A., Bremer, P.-T., Hamann, B., Pascucci, V.: Practical Considerations in Morse-Smale Complex Computation. In: Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications. Mathematics and Visualization, pp. 67–78. Springer (2011)
12. Gyulassy, A., Natarajan, V., Pascucci, V., Bremer, P.-T., Hamann, B.: Topology-Based Simplification for Feature Extraction from 3D Scalar Fields. In: Proceedings IEEE Visualization 2005, pp. 275–280. ACM Press (2005)
13. Gyulassy, A., Natarajan, V., Pascucci, V., Bremer, P.-T., Hamann, B.: A Topological Approach to Simplification of Three-Dimensional Scalar Functions. IEEE Transactions on Visualization and Computer Graphics 12(4), 474–484 (2006)
14. Gyulassy, A., Natarajan, V., Pascucci, V., Hamann, B.: Efficient Computation of Morse-Smale Complexes for Three-dimensional Scalar Functions. IEEE Transactions on Visualization and Computer Graphics 13(6), 1440–1447 (2007)
15. Matsumoto, Y.: An Introduction to Morse Theory. Translations of Mathematical Monographs, vol. 208. American Mathematical Society (2002)
16. Meyer, F.: Topographic distance and watershed lines. Signal Processing 38, 113–125 (1994)
17. Robins, V., Wood, P.J., Sheppard, A.P.: Theory and Algorithms for Constructing Discrete Morse Complexes from Grayscale Digital Images. IEEE Trans. Pattern Anal. Mach. Intell. 33(8), 1646–1658 (2011)
18. Roerdink, J., Meijster, A.: The Watershed Transform: Definitions, Algorithms, and Parallelization Strategies. Fundamenta Informaticae 41, 187–228 (2000)

Random Tessellations and Boolean Random Functions

Dominique Jeulin

Centre de Morphologie Mathématique,
Mathématiques et Systémes, 35 rue Saint-Honoré, 77300 Fontainebleau, France

Abstract. Generalizations of various random tessellation models generated by Poisson point processes are proposed and their functional probability $P(K)$ is given. They are interpreted as characteristics of Boolean random functions models, which provide a generic way of simulation of general random tessellations.

Keywords: Voronoi tessellation, random tessellation, Boolean random function.

1 Introduction

Some models of random tessellations in the Euclidean space R^n are defined from distances to the points x_k of a point process, usually the Poisson point process $\mathcal{P}$: the Voronoi tessellation is defined from the zones of influence of points x_k. Its generalizations like the Johnson-Mehl and the Laguerre tessellations use a time sequence of points, and a sequence of ponderations allocated to each point. It turns out that these models can be re-interpreted in the framework of Boolean random functions, with appropriate primary functions. In what follows, we propose new models of random tessellations based on local metrics attached to each point of the process. They correspond to specific Boolean random function models, which can be used for their simulation. The new models show a wide flexibility, generating tessellations with non planar boundaries, that can be used to simulate metallic grains [1] or foams [10].

2 Reminder on Random Tessellations

Random tessellations were formalized by G. Matheron in [11].

Definition 1. *Consider a locally compact denumberable space E and subsets C_i of E, belonging to $\wp(E)$. A tessellation Θ is a collection of classes $C_i \in \wp(E)$ with*

$$\cup_i C_i = E \quad \text{and} \quad C_i \cap C_j = \emptyset \quad \text{for} \quad i \neq j$$

We note $\Pi(E)$ (namely $\Pi_g(E)$) the set of tessellations of E (namely of tessellations with open (or point) classes). R_A is the subset of $\Pi_g(E)$, such that $A \subset E$ is contained in one class C.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 25–36, 2013.

Subsets R_G, where G are open parts of E, generate a σ algebra on $\Pi_g(E)$, $\sigma(R_G)$, on which a probability can be constructed. A random tessellation Θ is characterized by $P(R_G) = P\{G \subset C_i\}$. In the present paper, we study some models of random tessellations in the Euclidean space R^n. In addition, we will consider locally finite random tessellations, for which the random number of classes in every bounded domain D is a finite random number $N(D)$.

With this σ algebra, we can define events (and their probability) like "x belongs to a single class", "x_1 and x_2" belong to a single class, "x_1, x_2, ... x_m belong to k classes, or more generally "the compact set K is included in a single class". Note that the classes of a random tessellation in R^n can be split in several connected components, as is the case for the dead leaves tessellation [6], or for some tessellations introduced in this paper.

3 L_p Voronoi tessellations

3.1 Standard Voronoï Tessellation

Definition 2. *The Poisson Voronoï tessellation in the Euclidean space R^n is defined from zones of influence of Poisson points [4,16,17]. The class C_k of the tessellation containing point x_k of the Poisson point process $\mathcal{P}$ is defined by*

$$C_k = \{x \in R^n, d(x, x_k) < d(x, x_l), x_k \in \mathcal{P}, x_l \in \mathcal{P}, l \neq k\} \tag{1}$$

It is easy to show that every cell of the tessellation is an open set. Its closure is delimited by planar faces (planes in R^3 and segments in R^2) orthogonal to segments connecting neighbour points of $\mathcal{P}$. Indeed, using the Euclidean distance with

$$d^2(x, x_k) = \sum_{i=1}^{i=n} (x_i - x_{ki})^2$$

x_i being the coordinates of point x in R^n, the boundary separating cells C_k and C_l is obtained by means of

$$d^2(x, x_k) = d^2(x, x_l), \tag{2}$$

generating a linear equation with respects to coordinates x_i, which provides the equation of an hyperplane.

The physical interpretation of this model is the isotropic growth from random point germs. Two-phase models of materials were generated from 3D Voronoi tessellations in [7].

3.2 Anisotropic Voronoï Tessellation

A first change of the model is obtained by a non isotropic growth of germs. This can be made by using a Euclidean metric with positive eigenvalues λ_i and with orthogonal eigenvectors obtained by a rotation of the basis of vectors e_i.

In that case, the Euclidean metric is represented by a symmetric positive definite matrix $M(n,n)$. Noting X and X_k the vectors with coordinates x_i and x_{ki}, we get $d^2(x,x_k) = (X^t - X_k^t)M(X - X_k)$, noting X^t the transpose of vector X. Changing the Euclidean metric is equivalent to performing affine transformations in the directions of the eigenvectors, with ratios λ_i. Therefore, the resulting Voronoï tessellation is obtained by performing the corresponding affine transformations to the standard Voronoï tessellation, resulting in an anisotropic model, as considered in [15].

3.3 Use of the L_p Metric

Replacing the Euclidean metric by the L_p metric produces new models of tessellations. We have for the L_p metric with the integer p

$$d^p(x,x_k) = \sum_{i=1}^{i=n} |x_i - x_{ki}|^p \tag{3}$$

The separation between cells becomes

$$d^p(x,x_k) = d^p(x,x_l).$$

When $p > 1$, this expression gives polynomials with degree $p - 1$ with respect to coordinates. For $p = \{1,2\}$ the separations are planar. For $p = 3$, we get portions of quadrics. Increasing the value of p gives higher order polynomial surfaces. However the obtained tessellations are not isotropic in the Euclidean space, since the balls defined by $\sum_{i=1}^{i=n} |x_i|^p = r^p$ are not isotropic, except for $p = 2$, giving spheres. For $p = 1$, the balls are hypercubes with edges orthogonal to directions given by $(\pm 1, \pm 1, ..., \pm 1)$. When $p \to \infty$ we get the L_∞ metric, for which balls are hypercubes with edges parallel to the coordinates system. For $p = 1$ and $p = \infty$, the separations are parallel to the faces of the corresponding hypercubes.

3.4 Tesselations Defined from Local Metrics

To simulate locally anisotropic growth and local growth rates, it is interesting to start with a field of metrics depending on the location x. Depending on variations of the metric in space, tessellations with oriented cells following a field of orientations will be produced. This approach is followed by [8] in the context of meshing, using local Euclidean metrics, but no probabilistic properties is given.

Tesselations Defined with a Local Euclidean Metric. We will now attach to every Poisson point x_k the Euclidean metric defined by the matrix M_k. In general the matrices corresponding to different germs will be correlated.

The definition of the tessellation (2) becomes:

Definition 3. *The local Poisson Voronoï tessellation in the Euclidean space R^n is defined from zones of influence of Poisson points, using the Euclidean distance d_k for point x_k. The class C_k of the tessellation containing point x_k is defined by*

$$C_k = \{x \in R^n, d_k(x, x_k) < d_l(x, x_l), x_k \in \mathcal{P}, x_l \in \mathcal{P}, l \neq k\} \tag{4}$$

The separation between cells C_k and C_l is given by the equation

$$(X^t - X_k^t)M_k(X - X_k) = (X^t - X_l^t)M_l(X - X_l) \tag{5}$$

Rearranging the terms in equation (5), we get

$$X^t(M_k - M_l)X - 2X^t(M_kX_k - M_lX_l) + X_k^tM_kX_k - X_l^tM_lX_l = 0 \tag{6}$$

The separations of cells are made of portions of quadrics, and the edges are therefore portions of conics. Note that the cells C_k can be made of several connected components.

Tesselations Defined with a Local L_p Metric. We will now consider attached to each germ x_k a local L_p metric defined on a basis obtained from the orthonormal basis of R^n by a rotation matrix R_k and a system of positive weights a_{ki}. In this basis, the coordinates of point x become $X' = RX$ and expression (3) becomes

$$d_k^p(x, x_k) = \sum_{i=1}^{i=n} a_{ki} \left|x_i' - x_{ki}'\right|^p \tag{7}$$

The definition of the tessellation (2) becomes:

Definition 4. *The local Poisson Voronoï tessellation in the Euclidean space R^n is defined from zones of influence of Poisson points, using the L_p metric d_k for point x_k. The class C_k of the tessellation containing point x_k is defined by*

$$C_k = \{x \in R^n, d_k^p(x, x_k) < d_l^p(x, x_l), x_k \in \mathcal{P}, x_l \in \mathcal{P}, l \neq k\} \tag{8}$$

As before, cells C_k are not necessarily connected. Their separations are made of portions of hypersurfaces of degree p, and the edges are portions of curves of degree p.

3.5 Calculation of the Probability $P(K)$

Voronoï Tessellation with a Constant Metric L_p. Consider a compact set K. General expressions of the probability $P(K) = P(K \subset C_k)$ can be derived for the Voronoi models defined from a Poisson point process with intensity $\theta(x)$, by generalization of the results of Gilbert [4]. We note $B(x, r)$ the ball with center x and radius r, defined from the metric L_p. We have $B(x, r) = \{y, d^p(x, y) \leq r^p\}$

Theorem 1. *Consider a Voronoi tessellation of space defined from the Poisson point process with intensity $\theta(x)$, and the metric L_p. The probability $P(K) = P(K \subset C_k)$ is given by*

$$P(K) = \int_{R^n} \theta(dy) \exp -\theta(F(K,y)) \tag{9}$$

where $\theta(F(K,y)) = \int_{R^n} \theta(dx) 1_{F(K,y)}(x)$ is the measure of the Voronoi flower $F(K,y) = \cup_{x \in K} B(x, d(x,y))$. In the stationary case, for a constant intensity θ, equation (9) becomes

$$P(K) = \theta \int_{R^n} \exp -\theta \mu_n(F(K,y))dy, \tag{10}$$

μ_n being the Lebesgue measure in R^n.

Proof. We have $K \subset C_k \iff \forall x \in K$, $\forall l \neq k$, $d(x,x_k) \leq d(x,x_l) \Leftrightarrow$ the ball with center $x \in K$ and radius $d(x,x_k)$ contains no point of the Poisson point process. Calling $F(K,y) = \cup_{x \in K} B(x, d(x,y))$ the Voronoi flower [2] of K with center y, we have $K \subset C_k \Leftrightarrow F(K,x_k)$ contains no point of the process, with probability $\exp -\theta(F(K,x_k))$. Equation (9) is obtained by randomization of the point x_k, $\theta(dy)$ being the probability that the element of volume dy contains a point of the process.

When K is a connected compact set, $P(K)$ gives the probability for K to be included in a single connected component of the cell.

Tesselations Defined with a Local L_p Metric. We consider now random tessellations with local L_p metrics d_k^p and d_l^p, attached to germs x_k and x_l. These random metrics, defined by a set of random coefficients and a rotation, are independent for separate germs. They are characterized by some multivariate distribution function noted $\varphi(k)$. We note $B_l(x,r)$ the ball with center x and radius r, defined from the metric d_l^p. We have $B_l(x,r) = \{y, d_l^p(x,y) \leq r^p\}$. We call $F_{lk}(K,y) = \cup_{x \in K} B_l(x, d_k(x,y))$ the flower of K with center y, and metrics d_l^p and d_k^p.

Theorem 2. *Consider a random tessellation of space with local L_p metrics, defined from the Poisson point process with intensity $\theta(x)$. The probability $P(K) = P(K \subset C_k)$ is given by*

$$P(K) = \int_{R^n} \varphi(l)dl\varphi(k)dk\theta(dy) \exp -\theta(F_{lk}(K,y)) \tag{11}$$

In the stationary case, for a constant intensity θ, equation (11) becomes

$$P(K) = \theta \int_{R^n} \varphi(l)dl\varphi(k)dk \exp -(\theta \mu_n(F_{lk}(K,y)))dy \tag{12}$$

Proof. Conditionally to the metrics d_k^p and d_l^p, and to the location of a Poisson point x_k, we have $K \subset C_k \iff \forall x \in K$, $\forall l \neq k$, $d_k(x, x_k) \leq d_l(x, x_l)$. Therefore, for every point x of K, the ball $B_l(x, d_k(x, x_k))$ contains no point of the process, and finally $F_{lk}(K, x_k)$ contains no point of the process, with probability $\exp -\theta(F_{lk}(K, x_k))$. Equation (11) is obtained by randomization of the point x_k, followed by a randomization of the choice of metrics d_k^p and d_l^p.

It is possible to replace the deterministic intensity $\theta(x)$ by a realization of a positive random function, replacing the Poisson point process by a Cox process [3]. In that case, we obtain Cox based random tessellations. Their corresponding moments $P(K)$ are deduced from equations (9, 11) by taking their expectation with respect to the random intensity.

4 Extension to Johnson-Mehl and to Laguerre Random Tessellation

The Johnson-Mehl tessellation [14] is obtained by combining germination (through a sequential intensity $\theta(t)$) and growth (with growth rate $\alpha(t)$). The usual model is based on constant (with respect to time) germination (with intensity θ) and growth rate (with intensity α). During the time sequence, germs falling inside growing crystals are deleted. Considering the sequence of Poisson germs $\{x_k, t_k\}$, we have:

$$C_k = \{x \in R^n, d(x, (x_k, t_k)) + \alpha(t_k)t_k < d(x, (x_l, t_l)) + \alpha(t_l)t_l, \tag{13}$$

$$x_k \in \mathcal{P}, x_l \in \mathcal{P}, l \neq k\} \tag{14}$$

Extensions of this model are obtained by means of a L_p metric, instead of the Euclidean distance. We can also use a local metric in the process, to generate anisotropic growth:

$$C_k = \{x \in R^n, d_k(x, (x_k, t_k)) + \alpha(t_k)t_k < d_l(x, (x_l, t_l)) + \alpha(t_l)t_l, \tag{15}$$

$$x_k \in \mathcal{P}, x_l \in \mathcal{P}, l \neq k\} \tag{16}$$

The Laguerre tessellation [9,10] is a generalization of the Voronoi tessellation, where to each Poisson point x_k is attached a random radius R_k. The cell is now defined from the power $P(x, x_k) = d^2(x, x_k) - R_k^2$. We have:

$$C_k = \{x \in R^n, P(x, x_k) < P(x, x_l), x_k \in \mathcal{P}, x_l \in \mathcal{P}, l \neq k\} \tag{17}$$

Some germs x_l generate empty cells, depending on the values of R_k^2 and on the distance to other germs. Cells are bounded by portions of hyperplanes. New random tessellations can be defined, based on the L_p metric, replacing in equation (17) $P(x, x_k)$ by $d^p(x, x_k) - R_k^p$. In general, non planar cell separations will be generated.

An extension of the construction (8) to the local metric case is obtained if $P(x, x_k)$ and $P(x, x_l)$ are replaced by $d_k^p(x, x_k) - R_k^p$ and $d_l^p(x, x_k) - R_l^p$ in equation (17).

5 Random Tessellations and Boolean Random Functions

The previous constructions can be obtained as characteristics of some Boolean random functions, re-interpreting the definition of C_k in terms of distance function.

We will attach to every Poisson point x_k a primary random function $Z'_k(x)$ defined according to the distance used in various definitions (4, 8, 14, 16, 17). For instance, for the standard Voronoi model, the primary function is an increasing paraboloid of revolution, while for the extension to the local Euclidean metric, it is an increasing paraboloid with general ellipsoidal sections in R^n. For the Johnson-Mehl model, primary functions are cones (with ellipsoidal section in the local case for the Euclidean metric); for germ $\{x_k, t_k\}$, the primary function is translated upward by addition of the constant $\alpha(t_k)t_k$. Models based on the L_p metric make use of functions defined in R^n by $Z'_k(x) = \sum_{i=1}^{i=n} a_{ki} \left|x'_i\right|^p$. The distance function associated to a model built from Poisson germs is given by

$$Z(x) = \wedge_k Z'_k(x - x_k) \tag{18}$$

By definition the random function $Z(x)$ is an Infimum Boolean random function [6]. For the Johnson-Mehl model, the primary function becomes $Z'_k(x - x_k) + \alpha(t_k)t_k$. For the Laguerre tessellation model, it becomes $Z'_k(x - x_k) - R_k^p$.

Sections of primary functions at level z are balls defined by the corresponding metric. Define

$$B'_k(z) = \{x, Z'_k(x) < z\}$$

From equation (18) we have

$$B(z) = \{x, Z(x) < z\} = \cup_{x_k} B'_k(z)_{x_k} \tag{19}$$

By construction equation (19) $B(z)$ is a Boolean random set with convex primary grains $B'_k(z)$. Consider a compact set K and the infimum $Z_\wedge(K) = \wedge_{y \in K}\{Z(y)\}$. We have

$$P\{Z_\wedge(K) \geq z\} = \exp -\{E(\theta(B'_k(z) \oplus \check{K}))\} \tag{20}$$

and for the stationary case

$$P\{Z_\wedge(K) \geq z\} = \exp -\{\theta E(\mu_n(B'_k(z) \oplus \check{K}))\} \tag{21}$$

For the simulation of random tessellations, we just need to simulate realizations of the Boolean random function with primary functions Z'_k corresponding to the model. The boundaries of the tessellation are provided by the crest lines of the random functions, obtained by the watershed of the random function using as markers the Poisson points. By construction of the Boolean random functions, the location of crest lines, and therefore the boundaries of the classes of the resulting tessellation are invariant by a non decreasing transformation Φ (anamorphosis) of the values of $Z'_k(x)$ (for instance using $Z'^p_k(x)$ instead of $Z'_k(x)$), that is compatible with the order relationship, namely such that $z_1 < z_2$ implies $\Phi(z_1) < \Phi(z_2)$.

An alternative extraction of classes is given by their labels C_k. Starting from the simulation, and from the germs x_k, we generate in each point x a set of labels $L(x)$:

$$L(x) = \{k, Z(x) = Z'_k(x - x_k)\} \tag{22}$$

Points x with the single label k generate the interior of cell C_k. Points with two labels k and l are on the boundaries between cells C_k and C_l. In R^3, points with three labels are on the edges of the tessellation, and points with four labels are its vertices.

This is illustrated in Figure 1 by a simulation in R, where a non connected class C_k is generated by the point x_k. This is just obtained by application of equation (22), the distance to x_k of points located in the left part of C_k of the figure being shorter than the distance to other germs.

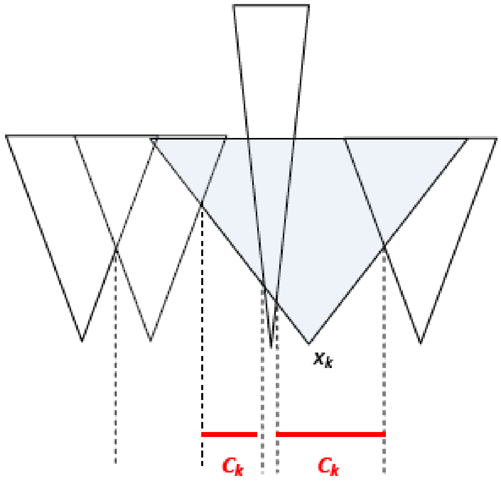


Fig. 1. Example of simulation of a random tessellation in R by means of a Boolean random function with different primary functions. The class C_k, generated by germ x_k, is not connected.

In Figure 2 is shown a realization of a Boolean random function BRF where the primary functions are doublets of elliptical cones with two orthogonal directions (the two vertical cones being obtained by a horizontal translation, and the two horizontal cones by a vertical translation) and the same minima. Figure 3 shows the corresponding local L_2 Voronoï random tessellation obtained from the watershed of the BRF. A simulation of a Boolean random function with vertial and horizontal elliptical cones having different minima is given in Figure 4. Its watershed in Figure 5 generates a realization of a local L_2 Johnson-Mehl random tessellation.

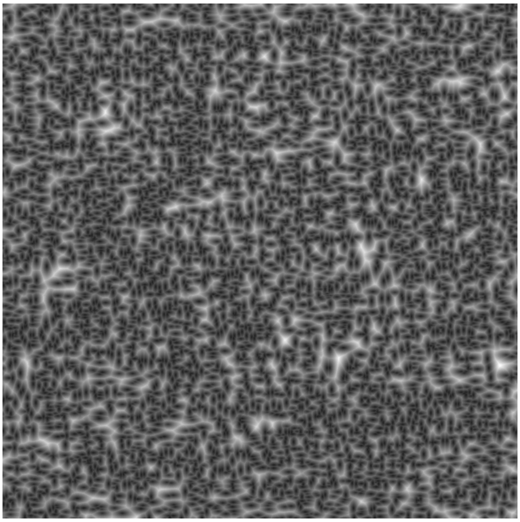


Fig. 2. Example of simulation of a Boolean random function with elliptical cones doublets in two orthogonal directions (image 800 x 800)

More general random tessellations can be generated by the same process, starting from Boolean random functions with any primary random function $Z'(x)$. We consider that the realization k of $Z'(x)$ is characterized by some

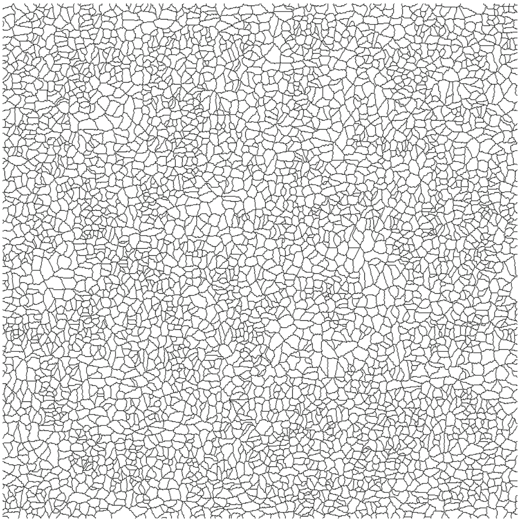


Fig. 3. Local L_2 Voronoï random tessellation generated by the realization of the Boolean random function of Figure 2 (image 800 x 800). The classes of the tessellation are obtained as the attraction zones of the minima of the BRF.

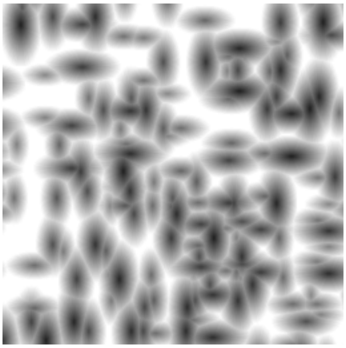


Fig. 4. Example of simulation of a Boolean random function with elliptical cones in two orthogonal directions. The primary functions start from different values (image 256 x 256).

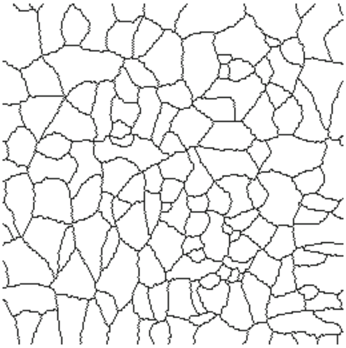


Fig. 5. Local L_2 Johnson-Mehl random tessellation generated by the realization of the Boolean random function of Figure 4 (image 256 x 256). The classes of the tessellation are obtained from the watershed of the BRF.

multivariate distribution $\varphi(k)$, and owns simply connected compact sections $B'_k(z)$, such that $B'_k(z_1) \subset B'_k(z_2)$ for $z_2 > z_1$. We consider primary random functions reaching their minimum $Z'(0)$ for $x = 0$. We associate to $Z'_k(x)$ the floor set A'_k defined by

$$A'_k = \{x, Z'_k(x) = Z'_k(0)\} \tag{23}$$

In all previous situations we had $A' = \{O\}$. If for any pair of Poisson points (x_k, x_l) we have $A'_{kx_k} \cap A'_{lx_l} = \varnothing$, we can define the class C_k of the random tessellation, generated by the germ x_k and the primary random function $Z'(x)$ by:

$$C_k = \{x \in R^n, Z'_k(x - x_k) < Z'_l(x - x_l), x_k \in \mathcal{P}, x_l \in \mathcal{P}, l \neq k\} \tag{24}$$

This construction of classes associated to germs works when $A' = \{O\}$. In R^2 it can also be applied when the floor set is made of parallel segments, while segments with two different orientations may overlap. In R^3, local anisotropy can be obtained by the Euclidean distance function to segments with different orientations, or even by Poisson lines [12,5] with an infinite length. In that case, point Poisson germs are replaced by segment germs or by lines. The generation of classes in simulations can be made by means of the previous procedure involving labels $L(x)$.

Generalizing the previous case of local metrics, we call $F_{lk}(K, y)$ the flower of K with center y, and primary functions $Z'_k(x)$ and $Z'_l(x)$. We have

$$F_{lk}(K, y) = \cup_{x \in K} B'_l(Z'_k(x - y))_x \tag{25}$$

The previous results are extended as follows.

Theorem 3. *Consider a random tessellation of space defined from the Poisson point process with intensity $\theta(x)$ and the primary random function $Z'(x)$ generating the flower defined by equation (25). The probability $P(K) = P(K \subset C_k)$ is given by*

$$P(K) = \int_{R^n} \varphi(l)dl\varphi(k)dk\theta(dy)\exp -\theta(F_{lk}(K,y)) \tag{26}$$

In the stationary case, for a constant intensity θ, equation (11) becomes

$$P(K) = \theta \int_{R^n} \varphi(l)dl\varphi(k)dk \exp -\theta\mu_n(F_{lk}(K,y))dy \tag{27}$$

Proof. Conditionally to the primary functions $Z'_k(x)$ and $Z'_l(x)$, and to the location of a Poisson point x_k, we have $K \subset C_k \iff \forall x \in K$, $\forall l \neq k$, $Z'_k(x-x_k) < Z'_l(x-x_l)$. Therefore, for every point x of K, the set $B'_l(Z'_k(x,x_k))_x$ contains no point of the process, and finally $F_{lk}(K,x_k)$ contains no point of the process, with probability $\exp -\theta(F_{lk}(K,x_k))$. Equation (11) is obtained by randomization of the point x_k, followed by a randomization of the choice of the primary functions $Z'_k(x)$ and $Z'_l(x)$.

As before, replacing the deterministic intensity $\theta(x)$ by a realization of a positive random function Θ, we obtain Cox based random tessellations with corresponding moments $P(K)$ deduced from equation (26) by taking its expectation with respect to the random intensity Θ.

6 Some Indications on Model Identification

For practical applications, the choice of a proper model has to be done from available information, usually 2D or 3D images of the microstructures. Several criteria can be used to select a representative model: the boundaries of cells can be fit to polynomials of degree p, given the order of the L_p metric. In a work on metallic grains fom EBSD images [1], pertinent information on local metrics is extracted from the inertia matrix of the grains. Finally, use should be made of the functional $P(K)$ computed from the equations or estimated from measurements on real or simulated microstructure. As a consequence of the presence of non connected classes generated by the models, the measurements on images should be restricted to connected compact sets K, directly obtained by erosion of the complementary set of the boundaries of classes by K.

7 Conclusion

Random tessellations models involving Poisson germs were revised and generalized by the use of local metrics attached to the germs. These models are related to particular Boolean random functions with particular primary functions generated by the metrics. Using other primary functions extends the type of random tessellations that can be simulated, and gives more flexibility to model complex real microstructures. Further generalizations of random tessellations can be obtained on Riemannian manifolds, equipped with a Riemannian metric, zones of influence of random points being generated by means of a geodesic distance, as already studied on the sphere [13]. Most results on the present study can be extended to this situation. In R^n, the L_p metrics can also be replaced by a geodesic distance, giving access to more general models.

References

1. Altendorf, H., Latourte, F., Saintoyant, L., Jeulin, D., Faessel, M.: Analysis and Stochastic Modeling of the Microstructures of Steel P91/P92 for Fracturation Studies at Welding Joints, submitted to ECS 11, Kaiserslautern (July 2013)
2. Calka, P.: Asymptotic methods for random tessellations. In: Spodarev, E. (ed.) Stochastic Geometry, Spatial Statistics and Random Fields. Lecture Notes in Mathematics, vol. 2068, ch. 6, Springer (2013)
3. Cox, D.R.: Some statistical methods connected with series of events. Journal of the Royal Statistical Society. Series B 17, 129–164 (1955)
4. Gilbert, E.N.: Random subdivisions of space into crystals. The Annals of Mathematical Statistics 33, 958–972 (1962)
5. Jeulin, D.: Modèles de Fonctions Aléatoires multivariables. Sci. Terre 30, 225–256 (1991)
6. Jeulin, D.: Modèles Morphologiques de Structures Aléatoires et de Changement d'Echelle. Thèse de Doctorat d'Etat ès Sciences Physiques, Université of Caen (1991)
7. Kanit, T., Forest, S., Galliet, I., Mounoury, V., Jeulin, D.: Determination of the size of the representative volume element for random composites: statistical and numerical approach. Int. J. Sol. Str. 40, 3647–3679 (2003)
8. Labelle, F., Shewchuk, J.R.: Anisotropic Voronoi Diagrams and Guaranteed-Quality Anisotropic Mesh Generation. In: Proceedings of the Nineteenth Annual Symposium on Computational Geometry, San Diego, pp. 191–200 (2003)
9. Lautensack, C., Zuyev, S.: Random Laguerre Tessellations. Advances in Applied Probability 40, 630–650 (2008)
10. Lautensack, C.: Fitting three-dimensional Laguerre tessellations to foam structures. Journal of Applied Statistics 35(9), 985–995 (2008)
11. Matheron, G.: Introduction à la théorie des ensembles aléatoires. Cahiers du Centre de Morphologie Mathématique, Fascicule 4 (1969)
12. Matheron, G.: Random sets and Integral Geometry. Wiley (1975)
13. Miles, R.E.: Random Points, Sets and Tessellations on the Surface of a Sphere, Sankhyā. The Indian Journal of Statistics, Series A, 145–174 (1971)
14. Møller, J.: Random Johnson-Mehl tessellations. Advances in Applied Probability 24, 814–844 (1992)
15. Scheike, T.H.: Anisotropic growth of Voronoi Cells. Advances in Applied Probability 26, 43–53 (1994)
16. Serra, J.: Image analysis and Mathematical Morphology. Academic Press, London (1982)
17. Stoyan, D., Kendall, W.S., Mecke, J.: Stochastic Geometry and its Applications. J. Wiley, New York (1987)

Discrete Set-Valued Continuity and Interpolation

Laurent Najman[1] and Thierry Géraud[2]

[1] Université Paris-Est, LIGM, Équipe A3SI, ESIEE
[2] EPITA Research and Development Laboratory (LRDE)
l.najman@esiee.fr, thierry.geraud@lrde.epita.fr

Abstract. The main question of this paper is to retrieve some continuity properties on (discrete) T0-Alexandroff spaces. One possible application, which will guide us, is the construction of the so-called "tree of shapes" (intuitively, the tree of level lines). This tree, which should allow to process maxima and minima in the same way, faces quite a number of theoretical difficulties that we propose to solve using set-valued analysis in a purely discrete setting. We also propose a way to interpret any function defined on a grid as a "continuous" function thanks to an interpolation scheme. The continuity properties are essential to obtain a quasi-linear algorithm for computing the tree of shapes in any dimension, which is exposed in a companion paper [10].

1 Introduction

This paper is the first in a series dedicated to the notion of the tree of shapes, which has been introduced [16, 7] as a way to filter an image $u : X \subset \mathbb{R}^n \to \mathbb{R}$ $(n \geq 2)$ in a self-dual way, meaning intuitively that processing either u or $-u$ would give the same result. In its continuous definition, this tree is made by the connected components of upper level sets $\{x; u(x) \geq \lambda\}_{\lambda \in \mathbb{R}}$ and lower level sets $\{x; u(x) < \lambda\}_{\lambda \in \mathbb{R}}$. This definition, by itself, is not really self-dual, as the tree of shapes of u is not the same as the tree of shapes of $-u$.

The goal of this paper is to propose a purely discrete framework in which a true self-dual definition of the tree of shapes can be given, together with a proof of the existence and uniqueness of such a tree for a given class of functions. We then provide several ways to interpret any image as belonging to this class of functions, one of them being self-dual. In the next paper of this series [10], we provide a quasi-linear algorithm relying on this theoretical framework, that allows the computation of the tree of shapes of an image, whatever its dimension.

In order to achieve our goal, we extend the notion of set-valued upper-semicontinuity [4] to the discrete case (T0-Alexandroff topology), and from which we build the notions of simple map and plain map that share the main properties of a classical continuous function. To obtain the tree of shapes, we also need to adapt the notion of well-composed map to our framework.

First approaches to discrete continuity date back to Rosenfeld [19], followed by Boxer [6]. Many authors have recognized that the set-valued setting is important to the discrete case [12, 22, 8]. None has proposed what we develop in this paper, although it may appear very natural for a researcher familiar with set-valued analysis.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 37–48, 2013.

2 Set-Valued Continuity on Discrete Spaces

2.1 Topology Reminder

A *topological space* is a set X composed of elements, or points, of arbitrary nature in which certain subsets $A \subseteq X$, called *closed sets* of the topological space X, have been defined so as to satisfy the following conditions, called the *axioms of a topological space*:

1. The intersection of any number and the union of any finite number of closed sets is a closed set.
2. The whole set X and the empty set $\emptyset$ are closed.

The sets complementary to the closed sets of X are called the *open sets* of the topological space X.

The intersection of all closed sets containing a set $M \subset X$ is called the *closure* of M in the topological space X and is denoted by $cl_X(M)$. Every open set containing a set M is called a *neighborhood* of the set M. A set $M \subset X$ is said to be *degenerate* if it contains just one point.

Definition 1. *A topological space X is said to be a* T_0-space *if every two distinct degenerate subsets of X have distinct closures in X. A T_0-space is called a* discrete space *if the union of an arbitrary number of closed sets of the space is closed.*

Finite T_0-spaces are the most important cases of the so called *discrete spaces* [1–3]. Another important example is the Khalimsky grid of dimension n, which is define as follows:

$$H_0^1 = \{\{x\}; x \in \mathbb{Z}\} \tag{1}$$
$$H_1^1 = \{\{x, x+1\}; x \in \mathbb{Z}\} \tag{2}$$
$$H^1 = H_0^1 \cup H_1^1 \tag{3}$$
$$H^n = \{h_1 \times \ldots \times h_n, \forall i \in [1,n], h_i \in H^1\} \tag{4}$$

In the Khalimsky grid, the elements of H_0^1 are closed sets, and the elements of H_1^1 are open sets. Although our definitions and results are valid in any discrete space, we will illustrate them on (a subset of) the Khalimsky grid.

If X is a discrete space, the intersection of all open sets containing a set $M \subset X$ is an open set, called the *star* of M in the topological space X. The star of M in X is denoted by $st_X(M)$. If $M = \{x\}$ is degenerate, by abuse of notation, we write $st_X(x) = st_X(\{x\})$.

Definition 2. *A set is said to be* connected *if it is not the union of two disjoint nonempty closed sets.*

For discrete spaces, this definition is equivalent to the one provided by the *connectivity by paths.* Two points x and y of X of a discrete space X are neighbors if either $y \in st_X(x)$ or $x \in st_X(y)$. A path in X from x to y is a sequence

$x_0 = x, x_1, \ldots, x_n = y$ such that x_{i+1} is a neighbor of x_i. One can prove that a set is connected if for any two points of this set, there exists a path between them. A *connected component* of X is a connected subset of X that is maximal for the connectivy property. We can also mention that any discrete space is locally connected, in the sense that any point $x \in X$ has a smallest open connected neighborhood in X, namely $st_X(x)$. The following lemma is useful in the sequel.

Definition 3. *Let M and N be two subspaces of X. M and N are* separated *if each is disjoint from the closure of the other, that is, if $(M \cap cl_X(N)) \cup (N \cap cl_X(M)) = \emptyset$.*

Lemma 4. *If two open sets of a discrete space are disjoint, then they are separated.*

2.2 Set-valued Maps

In the sequel, X and Y denotes two discrete spaces. The main references for set valued analysis is [4]. An application F is called a *set-valued map* from X to Y if for any $x \in X$, F associates to x the set $F(x) \subset Y$, called the *image* of F at x. In this case, the *domain* of F is the set $Dom(F) = \{x : F(x) \neq \emptyset\}$.

A set-valued map $F : X \rightsquigarrow Y$ is called *upper semicontinuous at $x \in Dom(F)$* (USC *at x*) if and only if for all $y \in st_X(x)$, $F(y) \subseteq st_Y(F(x))$. F is said to be upper semicontinuous if and only if it is upper semicontinuous at every point $x \in Dom(F)$.

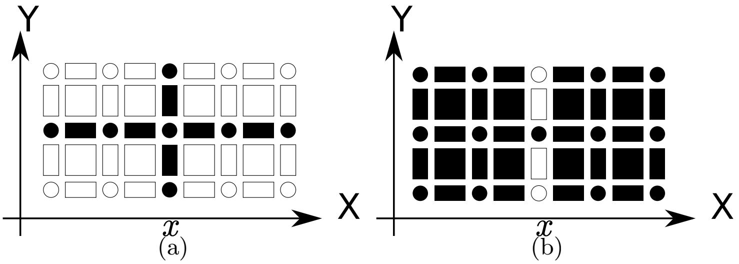


Fig. 1. (a) A discrete set-valued map F that is USC at x, drawn on a subset of the Khalimsky grid H_2 Note that F is also an interval-valued map, and x is both a minimum and a maximum of F. (b) A discrete set-valued map that is LSC at x.

One can remark that upper semicontinuity is the natural adaptation to set-valued map of the definition of a continuous function. An example of a USC set-value map is given in Fig. 1(a).

A useful property of a continuous function is that the inverse image of a closed (open) set is a closed (open) set. To adapt this property to set valued maps, we need to define the inverse. Let M be a subset of Y , $F : X \rightsquigarrow Y$

be a set-valued map. We denote $F^{\oplus}(M) = \{x \in X; F(x) \cap M \neq \emptyset\}$, and $F^{\ominus}(M) = \{x \in X; F(x) \subseteq M\}$. The subset $F^{\oplus}(M)$ is called the *inverse image* of M by F and $F^{\ominus}(M)$ is called the *core* of M by F.

Proposition 5. *A set-valued map $F : X \rightsquigarrow Y$ is* USC *if and only if the core of any open set is open: F is* USC *if and only if $F^{\ominus} st_Y = st_X F^{\ominus}$*

If furthermore F is with nonempty values (or if $Dom(F)$ is closed) F is USC *if and only if the inverse image of any closed subset is closed: F is* USC *if and only if $F^{\oplus} cl_Y = cl_X F^{\oplus}$.*

Intuitively, a function is continuous if, roughly speaking, it can be drawn as a single unbroken curve with no "holes" nor "jumps". In other words, a continuous function transforms a connected set into a connected set. For a set $M \subseteq X$, we write $F(M) = \cup_{x \in M} F(x)$.

Proposition 6. *Let F be a* USC *set-valued map such that, for all $x \in X$, $F(x)$ is a closed (*resp. *open) connected set. Then, for any connected set M, $F(M)$ is a closed (*resp. *open) connected set.*

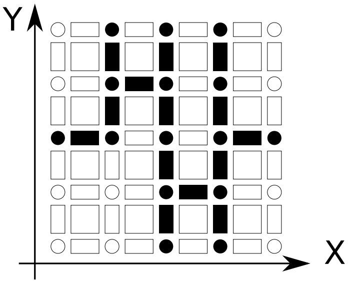


Fig. 2. A quasi-simple map

We say that a USC set-valued map F is a *closed (resp. open) quasi-simple map* if for all $x \in X$, $F(x)$ is a closed (resp. open) connected set, and if furthermore, for any $\{x\} = st_X(x) \in X$, $F(x)$ is degenerate. A quasi-simple map F is *simple* if it is the smallest of all quasi-simple maps with the same data on open points, *i.e*, a quasi-simple map F_1 is simple if for any quasi-simple map F_2 such that for any $\{x\} = st_X(x) \in X$, $F_1(x) = F_2(x)$, then, for all $x \in X$, $F_1(x) \subseteq F_2(x)$. Fig. 2 is an example of a quasi-simple map, while Fig. 3 is a simple map. The class of simple maps is interesting because it corresponds to our intuition (*i.e.* a single ubroken curve with no "holes" nor "jump") of continuous functions (at least when a simple map is interval-valued, see section 3).

In particular if f is a function defined for all $\{x\} = st_X(x) \in X$ with $f(x) \in Y$ and $f(x)$ closed (resp. open), there exists a (unique) closed (resp. open) simple map such that for all $\{x\} = st_X(x) \in X$, $F(x) = \{f(x)\}$.

Remark 1. V. Kovalevsky [12] defines the "continuous" functions as the ones that are called lower semicontuinous in our framework. More precisely, V. Kovalevsky uses prop. 8 below as the definition of a continuous function.

Definition 7. *A set-valued map $F : X \rightsquigarrow Y$ is called* lower semicontinuous *(*LSC*) at $x \in Dom(F)$ if for any open subset $M \subset Y$ such that $M \cap F(x) \neq \emptyset$, then for all $y \in st_X(x)$, $F(y) \cap M \neq \emptyset$.*

Fig. 1(b) gives an example of an LSC set-valued map. This definition is the adaptation to the (discrete) set-valued setting of the celebrated characterization of continuous function (a function f is continuous at x if and only if it maps any sequence converging to x to sequences converging to $f(x)$.) However, as no convergence can be defined in a discrete space, we have to use a topological definition. We have the following characterization of LSC maps.

Proposition 8. *A set-valued map $F : X \rightsquigarrow Y$ is* LSC *if and only if the inverse image of any open set is open: F is* LSC *if and only if $F^{\oplus} st_Y = st_X F^{\oplus}$*

If furthermore F is with nonempty values (or if $Dom(F)$ is closed) F is LSC *if and only if the core of any closed subset is closed: F is* LSC *if and only if $F^{\ominus} cl_Y = cl_X F^{\ominus}$.*

However, it is not possible to obtain simple LSC set-valued maps with interesting properties for our goal, and V. Kovalevsky then studies *connectivity-preserving* set-valued maps, that would be called *Darboux maps* in a more classical setting. In our framework (as in the classical one), some connectivity-preserving maps exist that are not USC, although prop. 6 shows that there exists USC maps preserving connectivity.

3 Interval-Valued Maps

Recall that a set X is said to be *unicoherent* if X is connected and for any two closed connected sets M and N of X such that $X = M \cup N$, then $M \cap N$ is connected. In the sequel, X_0 denotes a finite discrete topological space, and $X \subset X_0$ denotes an unicoherent subset of X_0.

An *interval* is a connected subset of H^1. Remark that H^1 is a totally ordered set, in the sense where H^1 is in bijection with $\frac{1}{2}\mathbb{Z}$: we identify the open sets of H^1 with the elements of $\frac{1}{2}\mathbb{Z} \setminus \mathbb{Z}$ and the closed sets of H^1 with the elements of $\mathbb{Z}$. We thus have $\ldots < \{0\} < \{0, 1\} < \{1\} < \ldots$, and operations such as $\lambda + \frac{1}{2}$ (where $\lambda \in H^1$), max or min on any subset of H^1 are well-defined. We say that a set-valued map F is an *interval-valued* map if the images of F are intervals. We say that an interval-valued map is *bounded* if there exists λ and μ in H^1 such that $\lambda \leq \min\{\nu \in F(X)\}$ and $\mu \geq \max\{\nu \in F(X)\}$. It is easy to check that interval-valued maps verify important classical theorems, such as for example, the *intermediate value theorem* (which is a simple consequence of Pr. 6 and of the connectedness of an interval).

As illustrated on Fig. 3, a simple closed-valued interval-valued map from H^1 to H^1 is what one would intuitively draw when dealing with a classical continuous function. Thus we will use the following definition.

Definition 9. *A closed-valued, interval-valued simple map F from X to H^1, with $dom(F) = X$ is called a* plain map *on X.*

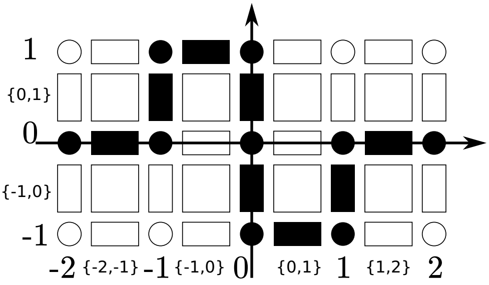


Fig. 3. A plain map F from H^1 to H^1

Remark that, as X is finite, a plain map is bounded. We have the following property.

Proposition 10. *If F is a plain map on X, then $F^\ominus$ is an open-valued* USC *map and $F^\oplus$ is a closed-valued* USC *map.*

3.1 Level Sets and Extrema

We need to adapt to the set-valued case the notions of upper and lower level sets. In the sequel, F denotes an interval-valued map from X to H^1. Let $\lambda \in \mathbb{Z}$, we write $[F \lhd \lambda] = F^\ominus(]-\infty, \lambda[) = \{\ x \in X \ \mid\ \forall \mu \in F(x),\ \mu < \lambda\ \}$. The set $[F \lhd \lambda]$ is called the *(strict) lower level set* (of F, at level λ). Similarly, the *(strict) upper level set* (of F, at level λ) is the set $[F \rhd \lambda] = F^\ominus(]\lambda, +\infty[)$. Thanks to prop. 10, the upper and lower level sets of a plain map are open.

The connected components of a set $M \subseteq X$ will be denoted by $\mathcal{CC}(M)$. If $x \in M$, the connected component of M that contains x will be denoted by $\mathcal{CC}(M, x)$, and by extension, we write $\mathcal{CC}(M, x) = \emptyset$ if $x \notin M$. If $\emptyset \neq C \subseteq M$ and C is connected, the connected component of M containing C, denoted by $\mathcal{CC}(M, C)$ is $\mathcal{CC}(M, x)$ with $x \in C$.

Definition 11 (Extrema). *A connected component of $[F \lhd \lambda]$ is a* minimum *of F if it does not contain any other connected component of $[F \lhd \mu]$ for any $\mu < \lambda$. A connected component of $[F \rhd \lambda]$ is a* maximum *of F if it does not contain any other connected component of $[F \rhd \mu]$ for any $\mu > \lambda$. An* extremum *of F is either a maximum or a minimum of F.*

We say that a set M is flat for F if for all $x \in M$, $F(x) = F(M)$. Remark that $F(M)$ is not always a degenerate set and that an extremum of a set-valued map is not always flat: for example, the point x in Fig. 1(a) is both a minimum and a maximum, and although this extremum is flat, it is not degenerate. It is easy to draw an example where a minimum of a set valued map contains several maxima, and hence is not flat.

Lemma 12. *The extrema of a plain map F are flat open sets. Furthermore, for any extrema M of F, $F(M)$ is degenerate.*

Proposition 13. *Let F be a plain map on X that is not constant. Then the extrema of F are separated.*

In the same way as the strict level sets, we can define the *(large) lower and upper level sets* $[F \trianglerighteq \lambda] = X \setminus [F \triangleleft \lambda] = F^{\oplus}(]-\infty, \lambda]) = \{ x \in X \mid \exists \mu \in F(x); \mu \leq \lambda \}$ and $[F \trianglelefteq \lambda] = X \setminus [F \triangleright \lambda]$. Thanks to prop. 10, $[F \trianglerighteq \lambda]$ and $[F \trianglelefteq \lambda]$ of a plain map F are closed.

Combining lower and upper level sets relations, we write $[F \equiv \lambda] = [F \trianglelefteq \lambda] \cap [F \trianglerighteq \lambda] = \{ x \in X \mid \lambda \in F(x) \}$ and $[F \not\equiv \lambda] = X \setminus [F \equiv \lambda] = \{ x \in X \mid \lambda \notin F(x) \} = [F \triangleleft \lambda] \cup [F \triangleright \lambda]$.

The following relations are natural. We indeed have $[F \trianglelefteq \lambda] = [F \triangleleft \lambda] \cup [F \equiv \lambda]$, $[F \trianglerighteq \lambda] = [F \triangleright \lambda] \cup [F \equiv \lambda]$ and $[F \not\equiv \lambda] = [F \triangleleft \lambda] \cup [F \triangleright \lambda]$. Similarly, we have for any λ_1 and λ_2 in H^1, $\lambda_1 < \lambda_2 \Rightarrow [F \trianglelefteq \lambda_1] \subseteq [F \trianglelefteq \lambda_2]$ and $\lambda_1 < \lambda_2 \Rightarrow [F \trianglerighteq \lambda_2] \subseteq [F \trianglerighteq \lambda_1]$. But if we have $\lambda_1 < \lambda_2 \Rightarrow [F \triangleleft \lambda_1] \cap [F \triangleright \lambda_2] = \emptyset$, the following relation is uncommon.

$$\lambda_1 < \lambda_2 \not\Rightarrow [F \trianglelefteq \lambda_1] \cap [F \trianglerighteq \lambda_2] = \emptyset \tag{5}$$

Indeed, for instance, for x such as $F(x) = [1, 2]$, we have both $x \in [F \trianglelefteq 1]$ and $x \in [F \trianglerighteq 2]$. Note that we may also have:

$$[F \triangleleft \lambda + \frac{1}{2}] \subsetneqq [F \trianglelefteq \lambda] \tag{6}$$

Indeed, with the same example, if $F(x) = [1, 2]$, $x \in [F \trianglelefteq 1]$, but $x \notin [F \triangleleft \frac{3}{2}]$. But, by the very definition of a plain map, if F is plain, for any $\lambda \in H^1$, we always have the following property.

Proposition 14. *A set-valued map F is a plain map if and only if, for any $\lambda \in H^1$,*

$$[F \trianglelefteq \lambda] = cl_X([F \triangleleft \lambda + \frac{1}{2}]) \text{ and } [F \trianglerighteq \lambda] = cl_X([F \triangleright \lambda - \frac{1}{2}]) \tag{7}$$

We can reconstruct F from its lower and upper level sets, with the following process: set $f_{\min}(x) = \min\{\lambda \in \mathbb{Z}; x \in [F \trianglelefteq \lambda]\}$ and $f_{\max}(x) = \max\{\lambda \in \mathbb{Z}; x \in [F \trianglerighteq \lambda]\}$. Then $F(x) = [f_{\min}(x), f_{\max}(x)]$. The process is simpler if F is plain: then for any degenerate open set $x \in X$, we have $F(x) = \{\min\{\lambda \in \mathbb{Z}; x \in [F \triangleleft (\lambda + 1)]\}\} = \{\max\{\lambda \in \mathbb{Z}; x \in [F \triangleright (\lambda - 1)]\}\}$. The other images of a simple/plain map F can be deduced from these ones.

Definition 15. *Let $\mathcal{T} \subseteq 2^X$. We say that $\mathcal{T}$ is a tree (of subsets of X) if (i) $X \in \mathcal{T}$, and (ii), if $M, N \in \mathcal{T}$, then either $M \cap N \neq \emptyset$, $M \subseteq N$, $N \subseteq M$. In the last two cases, we say that M and N are nested.*

For a plain map F, let us denote $\mathcal{U}^{\ominus}(F) = \{M; M \in \mathcal{CC}([F \triangleleft \lambda]), \lambda \in H^1\}$, $\mathcal{U}^{\oplus}(F) = \{M; M \in \mathcal{CC}([F \trianglelefteq \lambda]), \lambda \in H^1\}$, $\mathcal{L}^{\ominus}(F) = \{M; M \in \mathcal{CC}([F \triangleright \lambda]), \lambda \in H^1\}$ and $\mathcal{L}^{\oplus}(F) = \{M; M \in \mathcal{CC}([F \trianglerighteq \lambda]), \lambda \in H^1\}$. We remark that any two elements of one of these four sets are either disjoint or nested, hence they are trees. The set $\mathcal{U}^{\ominus}(F)$ (resp. $\mathcal{L}^{\ominus}(F)$) is the tree of open connected components of the lower (resp. upper) level sets of F, while $\mathcal{U}^{\oplus}(F)$ (resp. $\mathcal{L}^{\oplus}(F)$) is the tree of closed connected components of the lower (resp. upper) level sets of F. In the literature, similar trees have been called min- or max-trees, or component trees.

4 Shapes of Interval-Valued Maps

Topology is not so important when we consider independently the min or the max tree. It becomes fundamental when considering both trees at the same time, which is at the basis of the so-called tree of shapes.

Definition 16. *Let $M \subseteq X$. We call* cavities *of M in X the components of $X \setminus M$. Let $p_\infty \in X$ a reference point. We call* saturation *of M with respect to p_∞ the set $sat(M, p_\infty) = X \setminus \mathcal{CC}(X \setminus M, p_\infty)$.*

For any $\lambda \in H^1$, we call *quasi-shape* of x the set:

$$S_\lambda(F, x) \;=\; sat(\mathcal{CC}([F \not\ni \lambda], x), p_\infty) \tag{8}$$

We thus have two different *type*s possible for a given shape:

$$\begin{aligned} \text{either } S_\lambda(F, x) \;&=\; sat(\mathcal{CC}([F \rhd \lambda], x), p_\infty) & (9)\\ \text{or } S_\lambda(F, x) \;&=\; sat(\mathcal{CC}([F \lhd \lambda], x), p_\infty). & (10)\end{aligned}$$

But all the shapes being open, the type of a shape can not be known from the shape itself (contrary to the case of the continuous framework [7]).

Consider the family of sets:

$$\mathfrak{S}(F, x) \;=\; \{\, S_\lambda(F, x)\,\}_\lambda \setminus \emptyset. \tag{11}$$

and denote by $\mathfrak{S}(F)$ the set formed by all the $\mathfrak{S}(F, x)$ for all $x \in X$.

The notion of surface used in this paper is the one of n-surface.

Definition 17 (n-surface,[5, 9]). *A discrete space Y is a 0-surface if Y is made of exactly two points x and y such that $x \notin st_Y(y)$ and $y \notin st_Y(x)$. A discrete space Y is a n-surface ($n > 0$) if Y is connected and if, for any $x \in Y$, $cl_Y(x) \cup st_Y(x) \setminus \{x\}$ is a ($n-1$)-surface.*

The notion of well-composed connected set has been introduced in [13, 14]. Several extensions have been proposed in the literature [15, 11, 23]. In this paper, we extend it to maps by considering at the same time upper and lower level sets.

Definition 18 (Well-composed set and map). *A connected set M is* well-composed *if $\partial M = cl_X(M) \cap cl_X(X \setminus M)$ is a n-surface. A set N is* well-composed *if any connected component of ∂N is a n-surface. A plain map F is* well-composed *if for any $\lambda \in H^1$, both $[F \lhd \lambda]$ and $[F \rhd \lambda]$ are well-composed.*

Remark 2. Note that a well-composed set is a regular open set [18].

Lemma 19. *If a plain map F is well-composed, then for any $\lambda \in H^1$, and for any connected component $C \in \mathcal{CC}([F \trianglelefteq \lambda])$ (resp. $C \in \mathcal{CC}([F \trianglerighteq \lambda])$), there exists a connected component $CC \in \mathcal{CC}([F \triangleleft \lambda + \frac{1}{2}])$ (resp. $CC \in \mathcal{CC}([F \triangleright \lambda - \frac{1}{2}])$) such that $C = cl_X(CC)$.*

Lemma 20. *If a plain map F is well-composed, then for any $\lambda \in H^1$, and for any connected component $M \in \mathcal{CC}([F \trianglelefteq \lambda])$ (resp. $M \in \mathcal{CC}([F \trianglerighteq \lambda])$), we have $sat(cl_X(M)) = cl_X(sat(M))$.*

Theorem 21 *If F is a well-composed plain map, then any two quasi-shapes are either disjoint or nested. Hence $\mathfrak{S}(F)$ is a tree.*

The proof of Th. 21 is essentially the same as the one of Th. 2.6 in [7], adapted for the set-valued case of a plain map, and with a different ending that uses Lemma 20 and Lemma. 19. Due to space constraint, this proof will be provided in an extended version of this paper.

Definition 22. *We call* shape *of x the smallest non-empty quasi-shape of x defined by:*

$$S(F, x) = \bigcap_{S \in \mathfrak{S}(F,x)} S \tag{12}$$

We denote by $S(F)$ the set formed by all the shapes of x for all $x \in X$

The intersection in (12) is open as an intersection of open sets in a discrete space; moreover it is non-empty, as $S(F, x)$ is one of the quasi-shapes of x by proposition 21.

Corollary 23 *The set $S(F)$ is a tree (called the* tree of shapes*).*

5 Interpolation

In order to apply the framework, we need a way to interpret any function defined on a grid as a well-composed plain map. Unfortunately, directly interpreting a set of data as a plain map (as in Fig. 4.a) does not lead to a well-composed plain map. We do not want to modify the original data, as it is done in [20]. Closer to what we propose in this paper are the magnification process of [17], and even closer is [21], where new points are introduced thanks to subdivision of the space, the value on the points being derived from a majority rule. However, none of the previous approaches is equivalent to the simple interpolation we propose hereafter. More precisely, we map a function defined on $\mathbb{Z}^n$ to a function defined on the Khalimsky grid H^n using an adequate space subdivision.

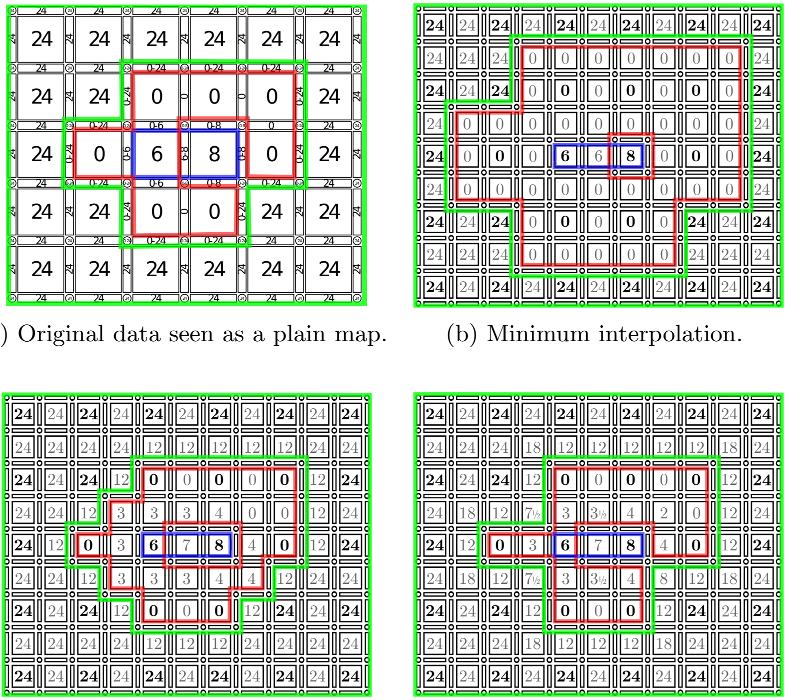
Let us denote by H_1^n the set of the elements of H^n that are open sets:

$$H_1^n = \{h_1 \times \ldots \times h_n; \forall i \in [1, n], h_i \in H_1^1\}.$$

With $2H_1^1 = \{\{2x, 2x+1\}; x \in \mathbb{Z}\}$, we can define several subsets of H_1^n:

$$2H_1^n = \{h_1 \times \ldots \times h_n; \forall i \in [1, n], h_i \in 2H_1^1\} \text{ and } I_1^n = H_1^n \setminus 2H_1^n$$

and the subset of I_1^n: $C_1^n = \{h_1 \times \ldots \times h_n; \forall i \in [1, n], h_i \in H_1^1 \setminus 2H_1^1\}$.



(a) Original data seen as a plain map.

(b) Minimum interpolation.

(c) Median interpolation.

(d) Mean interpolation.

Fig. 4. Original data (a) and (b,c,d) three different interpolations F of the same 2D image that are plain maps. The original image values are depicted in bold in (b,c,d); contours of connected components of $[F \lhd 7]$ are in red, whereas contours of connected components of $[F \rhd 5]$ are in green and blue. One can remark that (a) and (d) are not well-composed and that the saturation of the components of (a) and of (d) do not satisfy Lemma 19 and 20, contrary to the components of (b) and (c). In particular, the saturation of the red component in (a) or (d) does not include the blue component.

Let us denote by $\mathcal{M}(\mathbb{Z})$ the set of multisets having $\mathbb{Z}$ as the underlying set of elements; for instance we have $\{1,1,2\} \in \mathcal{M}(\mathbb{Z})$. Let op : $\mathcal{M}(\mathbb{Z}) \to \frac{1}{2}\mathbb{Z}$ be an operator over a multiset of integers that is increasing with respect to any element of the multiset, that is, $\forall z = \{z_1, \ldots, z_k\} \in \mathcal{M}(\mathbb{Z}), \forall i \in [1, k]$, we have:

$$z'_i \leq z_i \;\Rightarrow\; \mathrm{op}(\{z_1, \ldots, z_{i-1}, z'_i, z_{i+1}, \ldots, z_k\}) \leq \mathrm{op}(z).$$

Some well-known eligible operators are the minimum, the maximum, and the median (with $\mathrm{med}(\{a, b\}) = (a + b)/2$). From any operator op we can derive an "interpolation" of a function $f : \mathbb{Z}^n \to \mathbb{Z}$ into a plain map $F : H^n \rightsquigarrow H^1$:

$$\mathfrak{I}_{\mathrm{op}} : \begin{cases} (\mathbb{Z}^n \to \mathbb{Z}) \longrightarrow (H^n \rightsquigarrow H^1) \\ \qquad f \qquad\;\; \longmapsto \quad \mathfrak{I}_{\mathrm{op}}(f) \end{cases}$$

where $\mathfrak{I}_{\mathrm{op}}(f)$ is recursively defined by:

$$\forall h \in H^n,\ \mathfrak{I}_{\mathrm{op}}(f)(h) = \begin{cases} 2f(\frac{h_1}{2}, \ldots, \frac{h_n}{2}) & \text{if } h \in 2H_1^n \\ \mathrm{op}(\{\mathfrak{I}(f)(h');\ h' \in st_X(cl_X(h)) \cap 2H_1^n\}) & \text{if } h \in I_1^n \\ \mathrm{span}(\{\mathfrak{I}(f)(h');\ h' \in st_X(h) \cap H_1^n\}) & \text{otherwise.} \end{cases}$$

Furthermore, the elements of C_1^n shall not lead to extrema in $\mathfrak{I}_{\mathrm{op}}(f)$.

Proposition 24. *Any plain map obtained thanks to the max- or the min-interpolation is well-composed.*

The max- and min- interpolation reproduces the Rosenfeld 4- and 8- topology. The tree of shapes obtained with this interpolation is the one computed by all previous existing algorithms [7]. However, the max- or the min-interpolation do not lead to a self-dual filtering. This is illustrated in Fig 4.b.

Proposition 25. *The median interpolation of a function defined on $\mathbb{Z}^2$ leads to a self-dual plain map.*

This proposition is illustrated on Fig. 4.c.

Conjecture 26 *With the suitable space subdivision, the median interpolation leads to a self-dual map whatever the dimension of the space.*

Remark 27 *Fig. 4.d shows that the mean operator is not suited for our type of interpolation.*

6 Conclusion

In this paper, we have shown how to obtain a well-composed plain map from any set of data defined on a grid, and we provide a median-interpolation operator inducing a true self-dual tree. In another paper [10], we provide a quasi-linear algorithm for computing the tree of shapes from a n-dimensional well-composed plain map. A dedicated algorithm (less memory consuming) will also be provided for the 2D case.

Acknowledgements. This work received funding from the Agence Nationale de la Recherche, contract ANR-2010-BLAN-0205-03 and through “Programme d’Investissements d’Avenir” (LabEx BEZOUT n°ANR-10-LABX-58).

References

1. Alexandroff, P.: Diskrete Räume. Math. Sbornik 2(3), 501–518 (1937)
2. Alexandroff, P.: Combinatorial topology. Dover Publications (1947)
3. Arenas, F.: Alexandroff spaces. Acta Math. Univ. Comen. LXVIII(1), 17–25 (1999)
4. Aubin, J., Frankowska, H.: Set-valued analysis. Birkhauser (2008)

5. Bertrand, G., Couprie, M.: A model for digital topology. Discrete Geometry for Computer Imagery, 229–241 (1999)
6. Boxer, L.: Digitally continuous functions. PRL 15(8), 833–839 (1994)
7. Caselles, V., Monasse, P.: Geometric description of topographic maps. Lecture Notes in Mathematics, vol. 1984. Springer (2010)
8. Escribano, C., Giraldo, A., Sastre, M.A.: Digitally continuous multivalued functions, morphological operations and thinning algorithms. JMIV 42(1), 76–91 (2012)
9. Evako, A.: Dimension on discrete spaces. International Journal of Theoretical Physics 33, 1553–1568 (1996)
10. Géraud, T., Carlinet, E., Crozet, S., Najman, L.: A quasi-linear algorithm to compute the tree of shapes of nD images. In: Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 98–110. Springer, Heidelberg (2013)
11. Gonzalez-Diaz, R., Jimenez, M.-J., Medrano, B.: Well-composed cell complexes. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) DGCI 2011. LNCS, vol. 6607, pp. 153–162. Springer, Heidelberg (2011)
12. Kovalevsky, W.: A new concept for digital geometry. Shape in Pictures, NATO ASI Series F 126, 37–51 (1994)
13. Latecki, L., Eckhardt, U., Rosenfeld, A.: Well-composed sets. Computer Vision and Image Understanding 61(1), 70–83 (1995)
14. Latecki, L.: 3d well-composed pictures. CVGIP: Graphical Model and Image Processing 59(3), 164–172 (1997)
15. Marchadier, J., Arquès, D., Michelin, S.: Thinning grayscale well-composed images. PRL 25(5), 581–590 (2004)
16. Monasse, P., Guichard, F.: Fast computation of a contrast-invariant image representation. TIP 9(5), 860–872 (2000)
17. Nakamura, A.: Magnification in digital topology. In: Klette, R., Žunić, J. (eds.) IWCIA 2004. LNCS, vol. 3322, pp. 260–275. Springer, Heidelberg (2004)
18. Ronse, C.: Regular open or closed sets. Tech. Rep. Working Document WD59, Philips Research Lab., Brussels (1990)
19. Rosenfeld, A.: "Continuous" functions on digital pictures. PRL 4(3), 177–184 (1986)
20. Siqueira, M., Latecki, L.J., Tustison, N., Gallier, J., Gee, J.: Topological repairing of 3D digital images. JMIV 30(3), 249–274 (2008)
21. Stelldinger, P., Latecki, L., Siqueira, M.: Topological equivalence between a 3d object and the reconstruction of its digital image. PAMI 29(1), 126–140 (2007)
22. Tsaur, R., Smyth, M.B.: "Continuous" multifunctions in discrete spaces with applications to fixed point theory. In: Bertrand, G., Imiya, A., Klette, R. (eds.) Digital and Image Geometry. LNCS, vol. 2243, pp. 75–88. Springer, Heidelberg (2002)
23. Wang, Y., Bhattacharya, P.: Digital connectivity and extended well-composed sets for gray images. CVIU 68(3), 330–345 (1997)

Solving Problems in Mathematical Morphology through Reductions to the U-Curve Problem

Marcelo S. Reis and Junior Barrera

Institute of Mathematics and Statistics, University of São Paulo, Brazil
msreis@ime.usp.br,
http://www.ime.usp.br/~msreis

Abstract. The U-curve problem is an optimization problem that consists in, given a finite set S, a Boolean lattice $(\mathcal{P}(S), \subseteq)$ and a chain $\mathcal{L}$, minimize a function $c : \mathcal{P}(S) \to \mathcal{L}$ that satisfies an extension of Matheron's increasing-decreasing decomposition (i.e., a function that is decomposable in U-shaped curves). This problem may be used to model problems in the domain of Mathematical Morphology, for instance, morphological operator design and some types of combinatorial optimization problems. Recently, we introduced the U-Curve-Search (UCS) algorithm, which is a solver to the U-curve problem. In this paper, we recall the principles of the UCS algorithm, present a constrained version of Serra's formulation of the Tailor problem, prove that this problem is a U-curve problem, apply the UCS algorithm to solve it and compare the performance of UCS with another optimization algorithm. Besides, we present applications of UCS in the context of W-operator design.

Keywords: combinatorial optimization, morphological operator, W-operator, U-curve problem, Tailor problem.

1 Introduction

The U-curve problem is a type of combinatorial optimization problem that is characterized by a search space organized as a Boolean lattice and a cost function c (i.e., a function that takes values from elements of the Boolean lattice to a chain) that is decomposable in U-shaped curves (i.e., for any three elements A, B, C of the Boolean lattice, $A \subseteq B \subseteq C$ implies that $c(B) \leq max\{c(A), c(C)\}$). A decomposable in U-shaped curves cost function is an extension of Matheron's characterization of the class of set operators decomposable as the intersection of an increasing and a decreasing operator [1]. The U-curve problem was proposed in 2010 by Ris et al. [2]. Recently, we demonstrated that the U-curve problem is NP-hard [3].

Although the hardness of the U-curve problem, this problem has importance in the context of the W-operator estimation problem, since it may be used to model the window design step of morphological operator design (refer to Sec. 3.2). Besides, several lattice-based optimization problems (e.g., the subset sum problem)

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 49–60, 2013.

can be solved through a reduction to the U-curve problem (i.e., through an injection from the instances of a given lattice-based optimization problem to instances of the U-curve problem) [3].

In 2010, Ris et al. introduced an algorithm designed to tackle the U-curve problem [2]. Recently, we proposed the U-Curve-Search (UCS) algorithm, which is an improvement on the algorithm of Ris et al., to solve this problem [3]. UCS showed a better performance, in optimal and suboptimal experiments, with simulated and real data, when compared to a branch and bound, an exhaustive search, and the Sequential Forward Floating Search (SFFS) heuristic [3].

After this Introduction, in Sec. 2, we will present a formalization of the U-curve problem and the basic principles of the UCS algorithm. In Sec. 3, we will show two examples of application: solving an optimization problem related to the Tailor problem and the window design step of morphological operator design. Finally, in Conclusion section, we will summarize the results presented in this paper and point out possible future works.

2 The U-Curve Problem and the UCS Algorithm

In this section, we will present a formal definition of the U-curve problem and the basic principles of the U-Curve-Search (UCS) algorithm. The UCS algorithm, designed to tackle the U-curve problem, is an improvement on the algorithm proposed in 2010 by Ris et al. [3].

2.1 The U-curve Problem

Let S be a finite non-empty set. The power set $\mathcal{P}(S)$ is the collection of all subsets of S, including the empty set and S itself. A chain is a collection $\{X_1, X_2, \ldots, X_k\} \subseteq \mathcal{P}(S)$ such that $X_1 \subseteq X_2 \subseteq \ldots \subseteq X_k$. Let $\mathcal{X} \subseteq \mathcal{P}(S)$ be a chain and f be a function that takes values from $\mathcal{X}$ to a chain $\mathcal{L}$. f describes a U-shaped curve if, for any $X_1, X_2, X_3 \in \mathcal{X}$, then $X_1 \subseteq X_2 \subseteq X_3$ implies that $f(X_2) \leq max\{f(X_1), f(X_3)\}$.

A cost function c is a function defined from $\mathcal{P}(S)$ to $\mathcal{L}$. c is decomposable in U-shaped curves if, for each chain $\mathcal{X} \subseteq \mathcal{P}(S)$, the restriction of c to $\mathcal{X}$ describes a U-shaped curve. Let X be an element of $\mathcal{P}(S)$. If there does not exist another element Y of $\mathcal{P}(S)$ such that $c(Y) < c(X)$, then X is of minimum cost.

Now, let us define the central problem of this paper.

Problem 1. Given a non-empty set S and a cost function c that is decomposable in U-shaped curves, find an element of $\mathcal{P}(S)$ of minimum cost.

Problem 1 is known as the U-curve problem, and was proposed in 2010 by Ris et al. [2]. In the following, we will present the UCS algorithm, which is an optimal algorithm to solve the U-curve problem (i.e., it always returns an element of $\mathcal{P}(X)$ of minimum cost).

2.2 The UCS Algorithm

U-Curve-Search (UCS) is an optimal algorithm to tackle the U-curve problem. This algorithm has two important characteristics:

1. it manages the search space $\mathcal{P}(S)$ as a Boolean lattice $(\mathcal{P}(S), \subseteq)$ of degree $|S|$ such that: the smallest and largest elements are, respectively, $\emptyset$ and S, and the sum and product are, respectively, the usual union and intersection on sets;
2. and uses the evaluation of a cost function that is decomposable in U-shaped curves for search and pruning procedures.

The way UCS explores the search space is defined by these two characteristics. In the sequence, we will present how UCS manages the search space, visits elements and performs pruning.

Management of the Search Space. An interval $[A, B]$ of $\mathcal{P}(S)$ is defined as $[A, B] := \{X \in \mathcal{P}(S) : A \subseteq X \subseteq B\}$. Let $[\emptyset, L]$ be an interval with the leftmost term being the empty set. L is a lower restriction. In the same way, let $[U, S]$ be an interval with the rightmost term being the complete set. U is an upper restriction.

The UCS algorithm manages the search space $\mathcal{P}(S)$ through two collections of restrictions. The collection of lower (in the dual case, upper) restrictions defines a set of intervals that does not belong to the current search space (i.e., it defines the elements that were pruned from the search space). Hence, including an element X of $\mathcal{P}(S)$ in the collection of lower (upper) restrictions implies that every element Y in $\mathcal{P}(S)$ such that $Y \subseteq X$ ($X \subseteq Y$) does not belong to the current search space. In Fig. 1, we show the current search space of a Boolean lattice of degree 5 such that the collections of lower and upper restrictions are, respectively, $\{11100, 00001\}$ and $\{10010, 00001\}$; the elements of $\mathcal{P}(S)$ are represented by the characteristic vector in $\{0, 1\}^{|S|}$.

Search and Pruning. UCS starts a search selecting an element that belongs to the current search space; for instance, in Fig. 1, the element 01010 is in the current search space. Then, UCS computes the cost of the selected element X and verifies the cost of every element Y that belongs to the current search space and is adjacent to X in the graph of the Hasse diagram of the Boolean lattice $(\mathcal{P}(S), \subseteq)$; for instance, in Fig. 1, if $X = 01010$, then the element 00010 should have its cost verified, since it is adjacent to 01010 and belongs to the current search space. If an element Y has cost less or equal to $c(X)$, then Y becomes the X element of the next iteration; otherwise, the interval $[\emptyset, Y]$ (in the dual case, $[Y, S]$) is pruned from the search space. In Fig. 2, we show an example of a search iteration; UCS explores the elements that are adjacent to $X = 11100$: 11000 and 11110 do not belong to the search space; 10100 and 01100 belong to the search space, but both have cost greater than $c(11100)$, hence the intervals $[00000, 10100]$ and $[00000, 01100]$ are pruned from the search space; finally, 11101

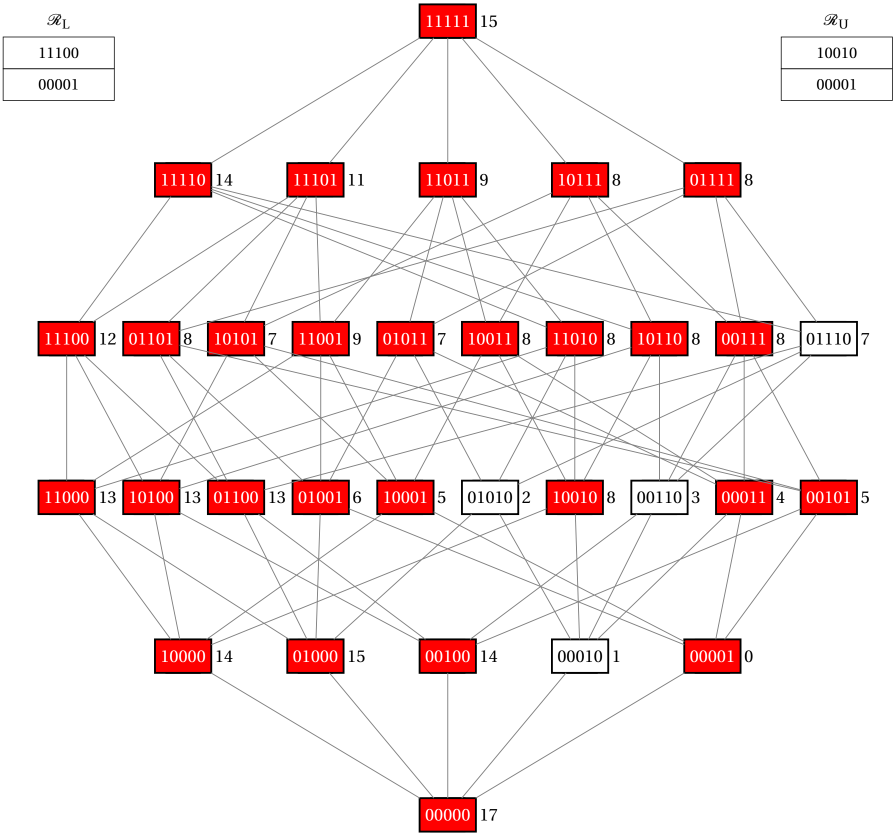


Fig. 1. A search space with $2^5 = 32$ elements. The current search space is composed by the elements assigned in white. The elements assigned in dark do not belong to the current search space because they are contained in one of the elements of the collection of lower restrictions ($\mathcal{R}_L$) and/or contain one of the elements of the collection of upper restrictions ($\mathcal{R}_U$); for instance, 10100 does not belong to the current search space because it is contained in 11100.

belongs to the search space and has cost smaller than $c(11100)$, hence it becomes X in the next iteration.

The pruning procedure takes into account a sufficient condition to remove elements from the search space without the risk of losing an element of minimum cost.

Proposition 1. *Let $\langle S, c\rangle$ be an instance of the U-curve problem, $\mathcal{P}(S)$ be the search space, $\mathcal{X} \subseteq \mathcal{P}(S)$ be a current search space, and X be an element of $\mathcal{X}$. If there exists an element $Y \in \mathcal{X}$ such that $Y \subseteq X$ and $c(Y) > c(X)$, then all elements in $[\emptyset, Y]$ have cost greater than $c(X)$.*

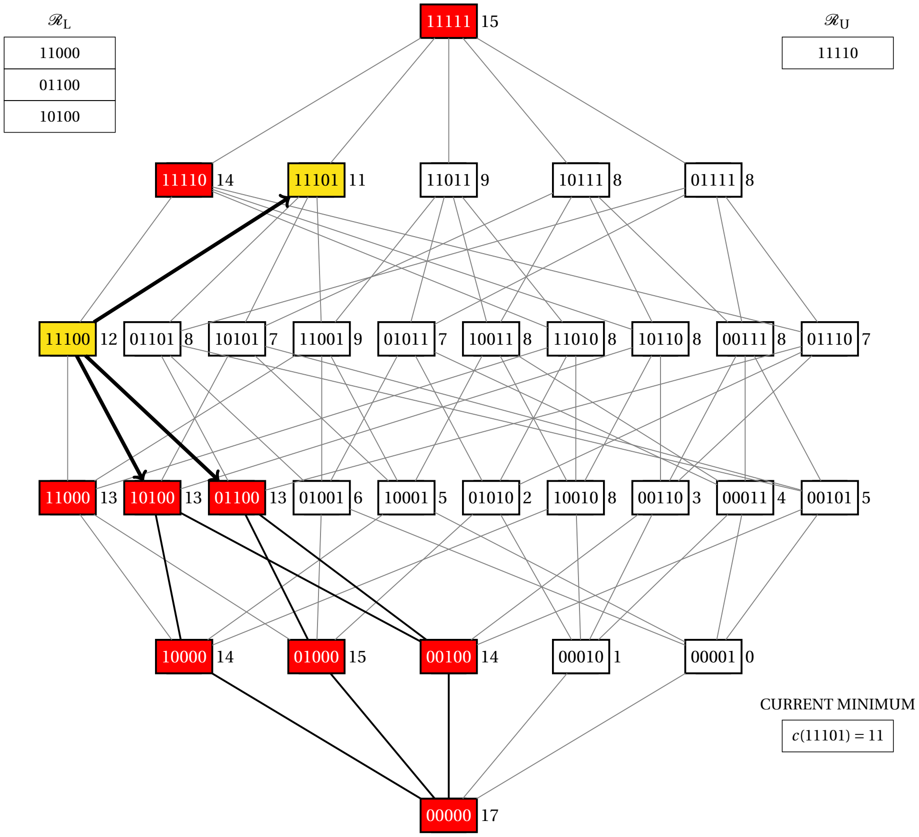


Fig. 2. Search and pruning procedures during an iteration of the UCS algorithm, on a search space with the same distribution of costs of the instance showed in Fig.1. The costs of the elements are assigned by the numbers beside the nodes. In this iteration, $X = 11100$ and the algorithm checks every element adjacent to 11100: 10100 and 01100 have cost greater than $c(11100)$, hence, by Proposition 1, the intervals $[00000, 10100]$ and $[00000, 01100]$ may be pruned from the search space. The element 11101 has cost smaller than $c(11100)$, thus it becomes X in the next iteration.

Proof. Let us consider $X, Y \in \mathcal{P}(S)$ and $Z \in [\emptyset, Y]$ such that $Y \subseteq X$ and $c(Y) > c(X)$. By the definition of a cost function decomposable in U-shaped curves, it holds that $c(Y) \leq max\{c(Z), c(X)\}$. Thus, $c(Y) \leq c(Z)$ or $c(Y) \leq c(X)$, and $c(Y) > c(X)$. Therefore, we have $c(Z) \geq c(Y) > c(X)$, which implies that all elements in $[\emptyset, Y]$ have cost greater than $c(X)$.

The result of Proposition 1 also holds for the Boolean lattice $(\mathcal{P}(S), \supseteq)$.

Proposition 2. *Let $\langle S, c\rangle$ be an instance of the U-curve problem, $\mathcal{P}(S)$ be the search space, $\mathcal{X} \subseteq \mathcal{P}(S)$ be a current search space, and X be an element of $\mathcal{X}$. If there exists an element $Y \in \mathcal{X}$ such that $X \supseteq Y$ and $c(X) > c(Y)$, then all elements in $[Y, S]$ have cost greater than $c(X)$.*

Proof. Applying the principle of duality, the result of Proposition 1 also holds for the Boolean lattice $(\mathcal{P}(S), \supseteq)$.

For a complete simulation of the UCS algorithm, as well as the employed data structures, pseudocodes of the main algorithm and its subroutines, proof of correctness and time complexity analysis, refer to Reis [3].

3 Application on Problems in Mathematical Morphology

We will present in this section two examples of application of the UCS algorithm on problems in Mathematical Morphology. The first one is a constrained version of the Tailor problem presented by Serra and the second one is an application to design window operators in the context of morphological operator design. These problems are U-curve problems, since their cost functions are defined on a Boolean lattice and are decomposable in U-shaped curves.

This section is organized in three parts: first, the presentation of the formulation of the Tailor optimization problem as a U-curve problem; second, the description of W-operator window design problem and its cost function decomposable in U-shaped curves; third, the experiments with instances of these two problems.

3.1 The Tailor Optimization Problem

In 1988, Serra proposed the Tailor problem (*le problème du Tailleur*) [4,5], which is defined as the following:

Problem 2. Given a set C and a collection of sets $P^1, \ldots, P^n$, decide if there is a collection of translations $P^1_{z_1}, \ldots, P^n_{z_n}$ such that:

- $(P^1_{z_1} \cup, \ldots, \cup P^n_{z_n}) \cap C^c = \emptyset$,
- and $P^i_{z_i} \cap P^j_{z_j} = \emptyset$, for all $i, j \in \{1, \ldots, n\}$, $i \neq j$.

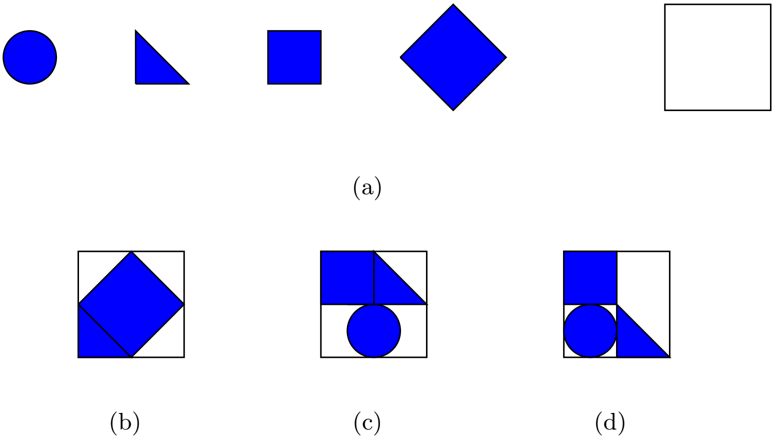
We will refer to the set C as "the cloth" and to the collection of sets $P^1, \ldots, P^n$ as "the pieces"; all those sets are subsets of $\mathbb{Z}^n$. If the problem has a solution, then we say that the pieces $P^1, \ldots, P^n$ fit in the cloth C.

Serra presented a solution for the Tailor problem in terms of Minkowski operations [5]. Although the Tailor problem is a decision problem, it is possible to use it to formulate related optimization problems: for instance, one might want to put as many pieces as possible into the cloth, and the arrangement of the pieces within the cloth must be one such that, for the maximum number of pieces that fit in C, it maximizes the greatest maximal convex set of remaining cloth (i.e., the elements of C that do not intersect with any piece). Therefore, we may define the following optimization problem.

Problem 3. Given a cloth C and a collection of pieces $P^1, \dots, P^n$, maximize the size k of a collection of translations $P^{i_1}_{z_{i_1}}, \dots, P^{i_k}_{z_{i_k}}$, $i_j \in \{1, \dots, n\}$ for all $1 \leq j \leq k$, such that:

- $P^{i_1}, \dots, P^{i_k}$ fit in C,
- and the maximal convex set in $C \cap (P^{i_1}_{z_{i_1}} \cup, \dots, \cup P^{i_k}_{z_{i_k}})^c$ must be maximum for all the collections of translations of size k that fit in C.

Problem 3 is known as the Tailor optimization problem. In Fig. 3, we present an instance of this problem, consisting in a cloth and four pieces, all of them subsets of $\mathbb{Z}^2$.


(a)

(b)

(c)

(d)

Fig. 3. An instance of the Tailor optimization problem. Fig. 3(a): the four pieces of the instance are assigned in dark, while the cloth C is the white-filled rectangle. Fig. 3(b): a set of two pieces that fit in C; the remaining cloth has three equivalent maximal convex sets. Fig 3(c): a set of three pieces that fit in C; this solution is better than the one of Fig. 3(b), since it has one more piece. Fig. 3(d): a set of three pieces that fit in C, translated in a way that the greatest maximal convex set of the remaining cloth is maximum; once it is not possible to fit in C a set of four pieces, this solution is optimal.

We will present now a formulation of the Tailor optimization problem as a U-curve optimization problem. Let $\mathcal{S}$ be the collection of all pieces $P_1, \dots, P_n$. The cost function c measures the subsets of $\mathcal{S}$, and it is defined as:

$$c(\mathcal{X}) = \begin{cases} -(|\mathcal{X}| + \frac{1}{|C|}|H_{\mathcal{X}}|), & \text{if the pieces in } \mathcal{X} \text{ fit in } C, \\ +\infty, & \text{otherwise}, \end{cases} \tag{1}$$

for all $\mathcal{X} \subseteq \mathcal{S}$, in which $H_{\mathcal{X}}$ is the greatest maximal convex set that is possible to obtain if the pieces in $\mathcal{X}$ fit in C. In the following result, we will demonstrate a property of the cost function c.

Theorem 1. *If $\mathcal{S}$ is a set of pieces, C is a cloth and c is a cost function defined as Eq. 1, then c is decomposable in U-shaped curves.*

Proof. Let $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{X}_3$ be elements of $\mathcal{P}(\mathcal{S})$ such that $\mathcal{X}_1 \subseteq \mathcal{X}_2 \subseteq \mathcal{X}_3$, and let c be a cost function defined as Eq. 1. Let us consider two cases.

In the first case, the pieces in $\mathcal{X}_2$ do not fit in C. Thus, we have that $c(\mathcal{X}_2) = +\infty$. Once $\mathcal{X}_2 \subseteq \mathcal{X}_3$, the pieces in $\mathcal{X}_3$ also do not fit in C, which implies that $c(\mathcal{X}_3) = +\infty$. Therefore, it holds that $c(\mathcal{X}_2) \leq max\{c(\mathcal{X}_1), c(\mathcal{X}_3)\}$.

In the second case, the pieces in $\mathcal{X}_2$ fit in C. If $\mathcal{X}_1 = \mathcal{X}_2$, then $c(\mathcal{X}_1) = c(\mathcal{X}_2)$, which implies that $c(\mathcal{X}_2) \leq max\{c(\mathcal{X}_1), c(\mathcal{X}_3)\}$; otherwise, $\mathcal{X}_1$ is properly contained in $\mathcal{X}_2$, and it holds that:

$$\begin{aligned}
c(\mathcal{X}_2) &= -(|\mathcal{X}_2| + \frac{1}{|C|}|H_{\mathcal{X}_2}|) && && (2)\\
&\leq -(|\mathcal{X}_1| + 1 + \frac{1}{|C|}|H_{\mathcal{X}_2}|) && \text{(since } |\mathcal{X}_1| < |\mathcal{X}_2|\text{)} && (3)\\
&\leq -(|\mathcal{X}_1| + 1) && \text{(since } |H_{\mathcal{X}_2}| \leq |C|\text{)} && (4)\\
&\leq -(|\mathcal{X}_1| + \frac{1}{|C|}|H_{\mathcal{X}_1}|) && \text{(since } |H_{\mathcal{X}_1}| \leq |C|\text{)} && (5)\\
&= c(\mathcal{X}_1) && \text{(by definition of } c\text{).} && (6)
\end{aligned}$$

Hence, we have that $c(\mathcal{X}_2) \leq max\{c(\mathcal{X}_1), c(\mathcal{X}_3)\}$. Therefore, we conclude that c is decomposable in U-shaped curves.

Once the cost function c is decomposable in U-shaped curves (Theorem 1), we can employ the UCS algorithm to find an element $\mathcal{X}$ of $\mathcal{P}(\mathcal{S})$ of minimum cost.

3.2 Design of Morphological Operators

We will show now an example of the application of the UCS algorithm on the window design step of the design of a morphological operator. More precisely, this example is a step of a W-operator design.

A W-operator is an image transformation that is locally defined inside a window and translation invariant [6]. A classifier characterizes an image transformation, and its design requires a window design procedure. The minimization of mean conditional entropy is a quality criterion that was showed effective in W-operator window design [7].

Let X be a subset of a window W and $\mathbf{X}$ be a random variable in $\mathcal{P}(X)$. The conditional entropy of a binary random variable Y given $\mathbf{X} = \mathbf{x}$ is given by

$$H(Y|\mathbf{X} = \mathbf{x}) = -\sum_{y \in Y} P(Y = y|\mathbf{X} = \mathbf{x}) \log P(Y = y|\mathbf{X} = \mathbf{x}), \quad (7)$$

in which $P(.)$ is the probability distribution function. By definition, $log 0 = 0$. The mean conditional entropy of Y given $\mathbf{X}$ is expressed by

$$E[H(Y|\mathbf{X})] = \sum_{\mathbf{x} \in \mathbf{X}} H(Y|\mathbf{X} = \mathbf{x}) P(\mathbf{X} = \mathbf{x}). \quad (8)$$

In practice, the values of the conditional probability distributions are estimated, thus a high error estimation may be caused by the lack of sufficient sample data (*i.e.*, rarely observed pairs $\langle y, \mathbf{x} \rangle$ may be underrepresented). In order to try to circumvent the insufficient number of samples, it is adopted the penalty for pairs of values that have a unique observation: it is considered a uniform distribution, thus leading to the highest entropy. Therefore, adopting the penalization, the estimation of the mean conditional entropy is given by

$$\hat{E}[H(Y|\mathbf{X})] = \frac{N}{t} + \sum_{\mathbf{x} \in \mathbf{X} : \hat{P}(\mathbf{x}) > \frac{1}{t}} \hat{H}(Y|\mathbf{X} = \mathbf{x}) \hat{P}(\mathbf{X} = \mathbf{x}), \tag{9}$$

in which N is the number of values of $\mathbf{X}$ with a single occurrence in the samples and t is the total number of samples.

Eq. 9 is a measure on estimated joint probability of feature subsets, which implies that it is susceptible to the U-curve phenomenon: for a fixed number of samples, the increase in the number of considered features may have two consequences: if the available sample is enough to a good estimation, then it should occur a reduction of the estimation error, otherwise, the lack of data induces an increase of the estimation error. Therefore, the U-curve problem may be employed to model the window design step, using X as the set of features and Eq. 9 as the cost function.

3.3 Experiments

In this section, we will present experimental results of the UCS algorithm in solving the Tailor optimization problem and performing the window design step of W-operator design. In these experiments, we compared the UCS algorithm with UBB, a branch and bound that was introduced by Ris et al. [3,2]. To execute these experiments, we used `featsel`, an object-oriented framework coded in C++ that allows implementation of solvers and cost functions over a Boolean lattice search space [8]. For the experiments showed in this paper, we implemented the UCS and the UBB algorithms. The experiments were done using a 32-bit PC with clock of 2.8 GHz and memory of 8 GB.

The Tailor Optimization Problem. In this experiment, Eq. 1 was employed as the cost function c; each time that c was computed, another branch-and-bound procedure was used to find an arrangement of the pieces within the cloth such that the greatest maximal convex set was maximum. This branch and bound enumerates the search space as a tree. For a collection of pieces $P^1, \ldots, P^k$, each level i of the tree, $1 \leq i \leq k$, represents trying to fit P^i in $C \cap (\cup_{j=1}^{i-1} P^j_{z_j})^c$ (i.e., a sequence of translations of the previous pieces that fit in C). If there is a fit of P^i, then the branch and bound goes to the next level, otherwise it backtracks. If there is a fit of the piece P^k, then the maximal convex set is computed.

Every test was done with a cloth of size 10 and the size of the pieces ranging from 1 to 3. For a test with a set of pieces $\mathcal{S}$ of size n, the instances were

Table 1. Comparison between the UCS algorithm and a branch and bound (UBB), using as the cost function Eq. 1, during the an execution of the Tailor optimization problem with a cloth of size 10 and the cardinality of the sets of pieces ranging from 5 to 14. The best performance, either in terms of required computational time or number of times that the cost function is computed, is underlined.

Size of instance		Time (seconds)		# Computed nodes	
$\vert\mathcal{S}\vert$	$2^{\vert\mathcal{S}\vert}$	UCS	UBB	UCS	UBB
5	32	0.0	0.0	6	32
6	64	0.1	0.2	19	64
7	128	0.3	0.5	20	128
8	256	0.8	1.6	41	255
9	512	2.8	4.8	155	506
10	1 024	11.4	17.5	308	1 006
11	2 048	30.1	41.5	837	1 917
12	4 096	183.6	282.6	1 642	3 797
13	8 192	440.9	563.0	3 592	7 108
14	16 384	1 071.5	1 319.4	7 571	13 643

obtained through the generation of n random pieces, with the number of pieces ranging from five to fourteen. For each size of instance, we produced ten different instances and carried out the two algorithms on each of them. Finally, for each size of instance, we took the average required time of the execution and the average number of computed nodes (i.e., the number of times the cost function was computed).

In Table 1, we summarize the results of this experiment. For instances from size 5 to 14, UCS always gave an optimal solution with a better ratio between size of instance / number of computed nodes than UBB; for example, for instances of size 14, the ratios of UCS and UBB were, respectively, 2.16 and 1.20. Besides, UCS always gave an optimal solution with a better ratio between size of instance / computational time than UBB; for example, for instances of size 14, the ratios of UCS and UBB were, respectively, 15.29 and 12.42.

Design of Morphological Operators. In this experiment, we employed the penalized mean conditional entropy (Eq. 9) as the cost function. The training samples were obtained from fourteen pairs of binary images presented in Martins-Jr et al. [7], running a 4×4 window of size 16 over each of observed image and obtaining the corresponding value of the transformed image. The result was fourteen sets of samples, each set with size 1152. For each set of samples, we executed the UCS algorithm and also the branch and bound described in Ris et al. [2]. We took the required time of the execution and the number of computed nodes (i.e., the number of times the cost function was computed).

In Table 2, we show the results of this experiment, in which each pair of images was used once for the window design procedure of the W-operator design.

Table 2. Comparison between the UCS algorithm and a branch and bound (UBB), using as the cost function a penalized mean conditional entropy, during a W-operator design with a window of size 16. The best performance, either in terms of required computational time or number of times that the cost function is computed, is underlined.

Test number	Time (seconds)		# Computed nodes	
	UCS	UBB	UCS	UBB
1	296.2	404.4	19 023	63 779
2	146.9	425.8	9 114	64 406
3	217.3	395.1	12 401	62 718
4	421.5	312.6	25 622	60 265
5	378.0	346.5	24 690	62 484
6	139.5	442.7	7 682	65 197
7	232.7	522.0	13 464	64 755
8	382.8	252.5	24 553	58 652
9	345.6	457.0	18 461	63 461
10	382.0	485.3	19 152	63 966
11	398.7	354.3	22 752	61 183
12	235.8	414.4	14 017	64 395
13	220.6	478.8	12 118	64 700
14	179.7	483.6	9 291	65 469
Average	284.1	412.5	16 595	63 245

UCS and UBB always gave an optimal solution. However, UCS gave an optimal solution in less computational time more frequently than UBB: on average, the required time (in seconds) of UCS and UBB were, respectively, 284.1 and 412.5. Additionally, UCS computed much less times the cost function than UBB: on average, the number of calls of the cost function of UCS and UBB were, respectively, 16595 and 63245.

4 Conclusion

In this paper, we presented a formalization of the U-curve problem and the basic principles of the U-Curve-Search (UCS) algorithm. UCS is an optimal algorithm to solve the U-curve problem. We showed that problems in Mathematical Morphology can be solved using the UCS algorithm. We gave two examples of application: solving the Tailor optimization problem and the window design step of morphological operator (namely a W-operator) design. In both examples, the UCS algorithm had a better performance than a branch and bound algorithm, either in terms of the required computational time or the number of times that the cost function was computed.

Other types of problems that the U-curve problem may also be employed include the minimization of the numbers of elementary morphological operators (i.e., erosion, dilation, anti-erosion, anti-dilation) required to represent a given morphological operator and, in the context of Pattern Recognition, the feature selection procedure of classifier design.

Future works in this line of research might include solve other problems in Mathematical Morphology using the UCS algorithm, further improvements on the UCS algorithm (especially in the way that the algorithm searches for an element of the current search space) and generalization of the UCS algorithm for non-Boolean lattices.

Acknowledgments. This work was supported by FAPESP project 08/51273−9 (Marcelo da Silva Reis), and by CNPq grant 306442/2011 – 6 and FAPESP Pronex 2011/50761 – 2 (Junior Barrera).

References

1. Banon, G.J.F., Barrera, J.: Minimal representations for translation-invariant set mappings by Mathematical Morphology. SIAM Journal on Applied Mathematics 51(6), 1782–1798 (1991)
2. Ris, M., Barrera, J., Martins Jr., D.C.: U-curve: A branch-and-bound optimization algorithm for U-shaped cost functions on Boolean lattices applied to the feature selection problem. Pattern Recogn. 43(3), 557–568 (2010)
3. Reis, M.S.: Minimization of decomposable in U-shaped curves functions defined on poset chains – algorithms and applications. PhD thesis, Institute of Mathematics and Statistics, University of São Paulo, Brazil (2012) (in Portuguese)
4. Serra, J.: Le problème du Tailleur. In: Int. report, CMM, School of Mines, Paris, pp. 1–5 (February 1988)
5. Roerdink, J.: The generalized tailor problem. Mathematical Morphology and its Applications to Image and Signal Processing, 57–64 (1996)
6. Barrera, J., Terada, R., Hirata Jr., R., Hirata, N.S.T.: Automatic programming of morphological machines by PAC learning. Fund. Inform., 229–258 (2000)
7. Martins Jr., D.C., Cesar Jr., R.M., Barrera, J.: W-operator window design by minimization of mean conditional entropy. Pattern Anal. Appl. 9(2), 139–153 (2006)
8. Reis, M.S.: featsel – a feature selection framework (2010), `http://www.ime.usp.br/~msreis/featsel`

Analytical Solutions for the Minkowski Addition Equation

Joel Edu Sánchez Castro, Ronaldo Fumio Hashimoto, and Junior Barrera

Institute of Mathematics and Statistics, University of Sao Paulo, Brazil
{edu,ronaldo,jb}@ime.usp.br

Abstract. This paper presents the formulation of a discrete equation whose solutions have a strong combinatory nature. More formally, given two subsets Y and C, we are interested in finding all subsets X that satisfy the equation (called *Minkowski Addition Equation*) $X \oplus C = Y$. One direct application of the solutions of this equation is that they can be used to find best representations for fast computation of erosions and dilations. The main (and original) result presented in this paper (which is a theoretical result) is an analytical solution formula for this equation. One important characteristic of this analytical formula is that all solutions (which can be in worst case exponential) are expressed in a compact representation.

Keywords: Minkowski Addition, Minkowski Subtraction, Estructuring Element Decomposition, Dilation, Erosion.

1 Introduction

Mathematical Morphology can be seen as a general framework for studying mappings between complete lattices. In particular, mappings between binary images are of special interest and they are called *set operators*. It is worth mentioning that set operators are widely used for binary image processing and analysis.

Two simple families of set operators are the so-called *dilations* and *erosions*. These set operators can be writen in terms of *Minkowski addition* and *subtraction*. More formally, if A and B are two subsets, the dilation (respectively, erosion) of A by B, denoted by $\delta_B(A)$ (respectively, $\varepsilon_B(A)$), is given by $\delta_B(A) = A \oplus B$ (respectively, $\varepsilon_B(A) = A \ominus B$). One important property of these set operators is that they are dual operators meaning that an erosion (or a Minkowski subtraction) can be written in terms of dilation (or Minkowski addition) and vice-versa. In this way, one can just consider to use one operator to obtain the other one.

In this paper, we present the formulation of a discrete equation whose solutions have a strong combinatory nature. More formally, given two nonempty subsets Y and C, we are interested in finding all subsets X that satisfy the equation (called *Minkowski Addition Equation*) $X \oplus C = Y$. One direct application of the solutions of this equation is that they can be used to find best representations for fast computation of erosions and dilations.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 61–72, 2013.

The main result presented in this paper (which is a theoretical result) is an analytical solution formula for this equation. One important characteristic of this formula is that all solutions (which can be in worst case exponential) are expressed in a compact form. Although there are some works that present some mathematical results on related problems in continuous domain (for example, for convex sets) [1,2], we did not find in the literature any work that finds all solutions for this equation and thereby the results given in this paper are original.

Following this introduction, Section 2 recalls some definitions and properties of binary Mathematical Morphology used in this paper. Section 3 introduces the Minkowski addition equation and, in Section 4, we present the analytical solution formula for this equation. Finally, Section 5 concludes the paper and give some future steps of this research.

2 Mathematical Foundations

This section provides the mathematical foundations necessary for presenting our main result.

Let E be a non-empty set and let $\mathcal{P}(E)$ denote the powerset of E.

The operations of *intersection* and *union* of subsets X and Y in $\mathcal{P}(E)$, are denoted, respectively, by $X \cap Y$ and $X \cup Y$.

Now, consider that the set E is also an Abelian group with respect to a binary operation denoted by $+$. Let $o \in E$ denote the origin (or zero element) of E. In this context, given a subset $X \in \mathcal{P}(E)$ and an element $h \in E$, the *translation* of X by h, denoted by $X+h$, is the set $X+h = \{x+h \in E : x \in X\}$. Furthermore, the *transpose* of X is the subset X^t given by $X^t = \{x \in E : -x \in X\}$.

Let us define two important binary operations in subsets. The *Minkowski addition and subtraction* of $X \in \mathcal{P}(E)$ and $Y \in \mathcal{P}(E)$ are, respectively, the subsets $X \oplus Y$ and $X \ominus Y$ given by $X \oplus Y = \bigcup\{X + h : h \in Y\}$ and $X \ominus Y = \bigcap\{X - h : h \in Y\}$.

Using these two binary operations, we can define two basic operators on subsets. Let C be a non-empty finite subset of E. The *dilation* and the *erosion* by C are the operators δ_C and ε_C given by, for any $X \in \mathcal{P}(E)$, $\delta_C(X) = X \oplus C$ and $\varepsilon_C(X) = X \ominus C$. The subset C is called a structuring element.

A property of dilations and erosions (or, equivalently, Minkowski additions and subtractions) is their sequential decomposability [3, p. 47].

Proposition 1. *Let $Y, C_1, C_2, \cdots, C_k$ be structuring elements. Then, we have that $\delta_Y = \delta_{C_1}\delta_{C_2}\cdots\ \delta_{C_k}$ and $\varepsilon_Y = \varepsilon_{C_1}\varepsilon_{C_2}\cdots\ \varepsilon_{C_k}$ if and only if $Y = C_1 \oplus C_2 \oplus\ \cdots\ \oplus\ C_k$.*

An immediate consequence of the last proposition is that, if Y can be decomposed into $Y = C_1 \oplus \cdots \oplus C_k$, then $X \oplus Y = X \oplus C_1 \oplus \cdots \oplus C_k$ and $X \ominus Y = X \ominus C_1 \ominus \cdots \ominus C_k$.

Such a decomposition of a structuring element (called *sequential decomposition*) has some nice properties. One of them is that the time complexity for dilation and erosion computation is extremely reduced. In fact, in sequential machines, the speed up was quantitatively studied by Maragos [4, p. 77], who showed examples where the time complexity went from quadratic to linear using sequential decomposition. Thus, several researchers [5,6,7,8,9,10,11] have studied the problem of decomposing structuring elements. In general, the time complexity is reduced since the total number of points in a sequential decomposition is less than the total number of the original structuring element. With this in mind, we can state another application: an error free compression when a subset is decomposable [12]. That is, this decomposition can be applied as an image compression or shape representation. Furthermore, this image representation, for example, can be used for image retrieval.

Now, let us state some other properties about Minkowski addition or subtraction that we will use in this paper. In the next proposition (proofs can be seen in [13]), we recall three important properties of Minkowski addition or subtraction.

Proposition 2. *Let* $X, Y, Z \in \mathcal{P}(E)$ *and* $h \in E$*, then*

1. $X \oplus Y = \{x + y \in E : x \in X, y \in Y\}$.
2. $X \subseteq Y \Rightarrow X \oplus Z \subseteq Y \oplus Z$.
3. $X \subseteq Y \Rightarrow X \ominus Z \subseteq Y \ominus Z$.

Minkowski addition and subtraction are dual in the sense that they satisfy the *adjunction relation* as stated in the following proposition (a proof can be found in [13]).

Proposition 3. *Let* $X, Y, C \in \mathcal{P}(E)$*,* $X \subseteq Y \ominus C$ *if and only if* $Y \supseteq X \oplus C$*.*

An important operation, called *morphological opening*, is a Minkowski subtraction followed by a Minkowski addition, that is, given $Y, C \in \mathcal{P}(E)$, the morphological opening of Y by C, denoted by $Y \circ C$, is $Y \circ C = Y \ominus C \oplus C$. A very important property of this operation (stated in Proposition 4) is that it is *anti-extensive*, in the sense that the morphological opening is always a subset of the original subset Y (for a proof, see, for example, reference [13]).

Proposition 4. *Let* $Y, C \in \mathcal{P}(E)$*, then* $Y \circ C = (Y \ominus C) \oplus C \subseteq Y$*.*

Of course, we may have cases where the morphological opening is a proper subset of the original subset. Now, when the morphological opening (that is, $Y \circ C$) and the original subset (that is, Y) happens to be equal (that is, $Y = (Y \ominus C) \oplus C$), we say that C is an *invariant* of Y.

An important property of invariant subsets, given in the next proposition, was stated by Zhuang and Haralick [11, Proposition 5].

Proposition 5. *Let* $C, X, Y \in \mathcal{P}(E)$*. If* $Y = X \oplus C$*, then* X *and* C *are both invariants of* Y*.*

3 Minkowski Addition Equation

In this paper, we introduce a formulation of a discrete equation whose solutions have a strong combinatory nature.

Problem 1. Given $Y, C \in \mathcal{P}(E)$, such that C and Y are nonempty sets, find all subsets $X \in \mathcal{P}(E)$ such that $X \oplus C = Y$.

At first, we should remark that, in some situations, this equation may have many solutions (in this sense, this problem is an ill-posed problem). In this way, Proposition 5 provides a search space reduction (that is, X must be an invariant of Y). In other situations, this equation may have no solution (when, for example, C is not an invariant of Y). Secondly, applying recursively the solutions of this equation, it is possible to find a sequential decomposition of Y (if one exists).

4 Analytical Solution for the Minkowski Addition Equation

In this section, we show an analytical formula that contains all solutions for the Minkowski equation. For that, Section 4.1 introduces an upper bound for all solutions of the equation; while Section 4.2 presents a collection of subsets that contains all the lower bounds for the solutions of the equation.

4.1 Upper Bound for Solutions

The following proposition (which is a direct consequence of Proposition 3) states an upper bound $U \in \mathcal{P}(E)$ for all solutions of the Minkowski equation, that is, if $X \in \mathcal{P}(E)$ is a solution of $X \oplus C = Y$, then $X \subseteq U$.

Proposition 6. *Let $Y \in \mathcal{P}(E)$ and $C \in \mathcal{P}(E)$. For all $X \in \mathcal{P}(E)$, if $X \oplus C = Y$, then $X \subseteq U$, where $U = Y \ominus C$.*

An important result (stated in the next proposition) is that if the Minkowski equation $X \oplus C = Y$ has solutions, then the upper bound $U = Y \ominus C$ must also be a solution.

Proposition 7. *Let $Y \in \mathcal{P}(E)$ and $C \in \mathcal{P}(E)$. If there exists $X \in \mathcal{P}(E)$ such that $X \oplus C = Y$, then $(Y \ominus C) \oplus C = Y$.*

Proof.

$$\begin{aligned}
X \oplus C = Y &\Rightarrow X \subseteq Y \ominus C && \text{(by Proposition 6)}\\
&\Rightarrow X \oplus C \subseteq (Y \ominus C) \oplus C && \text{(by Proposition 2)}\\
&\Rightarrow X \oplus C \subseteq (Y \ominus C) \oplus C \subseteq Y && \text{(by Proposition 4)}\\
&\Rightarrow Y \subseteq (Y \ominus C) \oplus C \subseteq Y && \text{(since } X \oplus C = Y\text{)}\\
&\Rightarrow (Y \ominus C) \oplus C = Y. && \square
\end{aligned}$$

4.2 Lower Bounds for Solutions

In this section, we build a set $\Phi^{Y,C} \subseteq \mathcal{P}(E)$ where, for each solution X of the Minkowski equation $X \oplus C = Y$, there exists an element $Z \in \Phi^{Y,C}$ such that $Z \subseteq X$. For that, we need the following result.

Proposition 8. *Let $C \in \mathcal{P}(E)$ and $Y \in \mathcal{P}(E)$. For all $X \in \mathcal{P}(E)$, if $X \oplus C = Y$, then for all $y \in Y$, there exist $x \in X$ and $c \in C$ such that $x + c = y$.*

Proof.

$$\begin{aligned} y \in Y &\Leftrightarrow y \in X \oplus C && \text{(since } X \oplus C = Y\text{)}\\ &\Leftrightarrow y \in \{x + c \in E : x \in X, c \in C\} && \text{(by Proposition 2)}\\ &\Leftrightarrow \exists x \in X, \exists c \in C : y = x + c. && \square \end{aligned}$$

Now, in order to build the set of lower bounds $\Phi^{Y,C}$, we need the following definition. Let $Y \in \mathcal{P}(E)$, $C \in \mathcal{P}(E)$ and $y \in E$. We define $L^{Y,C}(y)$ as:

$$L^{Y,C}(y) = \{x \in Y \ominus C : x = y - c, c \in C\}. \tag{1}$$

Let us give a simple example for the construction of the sets $L^{Y,C}(y)$. But, let us first introduce a notation for representing subsets in $\mathcal{P}(E)$. We represent a subset of E by an array of 0's and 1's, where '0' means that the point does not belong to the subset, and '1' means that the point belongs to the subset. Moreover, the origin $o \in E$ is represented by an underline character (such as '$\underline{0}$' or '$\underline{1}$') and this representation must be present in the array in order to be the reference for the other points. For example, if $E = \mathbb{Z}^2$, the subset $\{(-2, 0), (-1, 0), (1, 0), (1, 1), (1, -1)\}$ is represented by

$$\begin{array}{cccc} \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{1}\\ \mathtt{1} & \mathtt{1} & \underline{\mathtt{0}} & \mathtt{1}\\ \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{1}. \end{array}$$

Now, let us proceed with an example of $L^{Y,C}(y)$ sets construction.

Example 1. Let $Y = \mathtt{111\underline{1}11}$ and $C = \mathtt{1\underline{1}}$. Then $U = Y \ominus C = \mathtt{011\underline{1}11}$. Consider the following subsets (with just one element) $\{y_0\} = \mathtt{100\underline{0}00}$, $\{y_1\} = \mathtt{010\underline{0}00}$, $\{y_2\} = \mathtt{001\underline{0}00}$, $\{y_3\} = \mathtt{000\underline{1}00}$, $\{y_4\} = \mathtt{000\underline{0}10}$, $\{y_5\} = \mathtt{000\underline{0}01}$. Note that, $Y = \{y_0, y_1, y_2, y_3, y_4, y_5\}$. Then, after some little calculations, $L^{Y,C}(y_0) = \mathtt{10\underline{0}00} = \{y_1\}$, $L^{Y,C}(y_1) = \mathtt{11\underline{0}00} = \{y_1, y_2\}$, $L^{Y,C}(y_2) = \mathtt{01\underline{1}00} = \{y_2, y_3\}$, $L^{Y,C}(y_3) = \mathtt{00\underline{1}10} = \{y_3, y_4\}$, $L^{Y,C}(y_4) = \mathtt{00\underline{0}11} = \{y_4, y_5\}$, $L^{Y,C}(y_5) = \mathtt{00\underline{0}01} = \{y_5\}$. □

We can see that, by construction (see Eq. 1), $L^{Y,C}(y) \subseteq Y \ominus C$. Let us state one more property for $L^{Y,C}(y)$.

Proposition 9. *Let $C \in \mathcal{P}(E)$ and $Y \in \mathcal{P}(E)$. If there exists $X \in \mathcal{P}(E)$ such that $X \oplus C = Y$, then for all $y \in Y$, $L^{Y,C}(y) \neq \emptyset$.*

Proof.

$$\begin{aligned} X \oplus C = Y &\Rightarrow (Y \ominus C) \oplus C = Y && \text{(by Proposition 7)} \\ &\Rightarrow \forall y \in Y, \exists x \in Y \ominus C, \exists c \in C, x + c = y && \text{(by Proposition 8)} \\ &\Rightarrow \forall y \in Y, \exists x \in Y \ominus C, \exists c \in C, x = y - c && \text{(by addition in } E\text{)} \\ &\Rightarrow \forall y \in Y, \exists x \in L^{Y,C}(y) && \text{(by the definition of } L^{Y,C}(y)\text{)} \\ &\Rightarrow \forall y \in Y, L^{Y,C}(y) \neq \emptyset. && \square \end{aligned}$$

Using $L^{Y,C}(y)$, we can define a set that contains all the lower bounds for the solutions of the Minkowski equation. But, let us first define the cartesian product of two or more sets. Let $A_1, A_2, \ldots, A_n$ be subsets of $\mathcal{P}(E)$. The cartesian product of $A_1, \ldots, A_n$, denoted by $\prod_{i=1}^{n} A_i$, is a set of all n-tuples $(a_1, a_2, \ldots, a_n)$ such that $a_i \in A_i$, for all $i = 1, 2, \ldots, n$. More formally,

$$\prod_{i=1}^{n} A_i = A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \ldots, a_n) : a_i \in A_i, i = 1, 2, \ldots, n\}. \quad (2)$$

Now, we define the set $\Phi^{Y,C} \subseteq \mathcal{P}(E)$ (that contains all the lower bounds for the solutions of the Minkowski equation $X \oplus C = Y$ – see Theorem 2) as:

$$\Phi^{Y,C} = \{Z \in \mathcal{P}(E) : Z = \bigcup_{i=1}^{|Y|} \{x_i\}, (x_1, x_2, \ldots, x_{|Y|}) \in \prod_{y \in Y} L^{Y,C}(y)\}. \quad (3)$$

Let us give an example for the construction of $\Phi^{Y,C}$.

Example 2. For this example, consider the subsets described in Example 1. Let us first construct the set $\prod_{y \in Y} L^{Y,C}(y) = L^{Y,C}(y_0) \times L^{Y,C}(y_1) \times L^{Y,C}(y_2) \times L^{Y,C}(y_3) \times L^{Y,C}(y_4) \times L^{Y,C}(y_5)$. After some calculations, we have that $\prod_{y \in Y} L^{Y,C}(y)$ is composed by 16 ordered 6-tuples.

$$\begin{aligned} \prod_{y \in Y} L^{Y,C}(y) = \{ \; & (y_1, y_1, y_2, y_3, y_4, y_5), (y_1, y_1, y_2, y_3, y_5, y_5), \\ & (y_1, y_1, y_2, y_4, y_4, y_5), (y_1, y_1, y_2, y_4, y_5, y_5), \\ & (y_1, y_1, y_3, y_3, y_4, y_5), (y_1, y_1, y_3, y_3, y_5, y_5), \\ & (y_1, y_1, y_3, y_4, y_4, y_5), (y_1, y_1, y_3, y_4, y_5, y_5), \\ & (y_1, y_2, y_2, y_3, y_4, y_5), (y_1, y_2, y_2, y_3, y_5, y_5), \\ & (y_1, y_2, y_2, y_4, y_4, y_5), (y_1, y_2, y_2, y_4, y_5, y_5), \\ & (y_1, y_2, y_3, y_3, y_4, y_5), (y_1, y_2, y_3, y_3, y_5, y_5), \\ & (y_1, y_2, y_3, y_4, y_4, y_5), (y_1, y_2, y_3, y_4, y_5, y_5) \; \}. \end{aligned}$$

Thus, since $\Phi^{Y,C}$ is a collection of subsets which are obtained by the union of all elements in each ordered tuple, we have, after removing duplicate subsets,

$$\begin{aligned}\Phi^{Y,C} = \{ \; & \{y_1, y_2, y_4, y_5\}, \{y_1, y_3, y_4, y_5\}, \{y_1, y_3, y_5\}, \\ & \{y_1, y_2, y_3, y_5\}, \{y_1, y_2, y_3, y_4, y_5\} \; \}.\end{aligned}$$

In array representation, we have

$$\Phi^{Y,C} = \{ \; \texttt{11}\underline{\texttt{0}}\texttt{11},\ \texttt{10}\underline{\texttt{1}}\texttt{11},\ \texttt{11}\underline{\texttt{1}}\texttt{01},\ \texttt{10}\underline{\texttt{1}}\texttt{01},\ \texttt{11}\underline{\texttt{1}}\texttt{11} \; \}. \qquad \square$$

We observe that $|\Phi^{Y,C}| \leq \prod_{y \in Y} |L^{Y,C}(y)| \leq |C|^{|Y|}$. This indicates that the worst case time complexity to compute $\Phi^{Y,C}$ is an exponential function of $|Y|$.

Another observation (and also an important property) is that if $\Phi^{Y,C} \neq \emptyset$, then all elements in $\Phi^{Y,C}$ are solutions for the Minkowski equation.

Theorem 1. *Let $C \in \mathcal{P}(E)$ and $Y \in \mathcal{P}(E)$. If exists $X \in \mathcal{P}(E)$ such that $X \oplus C = Y$, then for all $Z \in \Phi^{Y,C}$, $Z \oplus C = Y$.*

Proof. On one hand, we prove that for all $Z \in \Phi^{Y,C}$, $Z \oplus C \supseteq Y$.

$$\begin{aligned}
Z \in \Phi^{Y,C} & \Rightarrow \exists (x_1, x_2, \ldots, x_{|Y|}) \in \prod_{y \in Y} L^{Y,C}(y) : Z = \bigcup_{i=1}^{|Y|} \{x_i\} \\
& \hspace{14em} \text{(by the definition of } \Phi^{Y,C}\text{)} \\
& \Rightarrow \forall y \in Y, \exists x \in L^{Y,C}(y) : x \in Z \quad \text{(by the union of sets in } Z \text{ and} \ldots\text{)} \\
& \hspace{10em} (\ldots \; \forall y \in Y, L^{Y,C}(y) \neq \emptyset \text{ – see Proposition 9)} \\
& \Rightarrow \forall y \in Y, \exists x \in Z, \exists c \in C : x = y - c \\
& \hspace{14em} \text{(by the definition of } L^{Y,C}(y)\text{)} \\
& \Rightarrow \forall y \in Y, \exists x \in Z, \exists c \in C : y = x + c \qquad \text{(by the addition in } E\text{)} \\
& \Rightarrow \forall y \in Y, \exists x \in Z : y \in \{x\} \oplus C \qquad \text{(since } c \in C\text{)} \\
& \Rightarrow \forall y \in Y, y \in Z \oplus C \qquad \text{(since } x \in Z\text{)} \\
& \Rightarrow Y \subseteq Z \oplus C. \qquad \text{(by the definition of subset)}
\end{aligned}$$

On the other hand, we prove that, for all $Z \in \Phi^{Y,C}$, $Z \oplus C \subseteq Y$.

$$
\begin{aligned}
Z \in \Phi^{Y,C} &\Rightarrow \exists (x_1, x_2, \ldots, x_{|Y|}) \in \prod_{y \in Y} L^{Y,C}(y) : Z = \bigcup_{i=1}^{|Y|} \{x_i\} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by the definition of } \Phi^{Y,C}\text{)} \\
&\Rightarrow \forall x \in Z, \exists y \in Y : x \in L^{Y,C}(y) \qquad\qquad \text{(by the definition of Z)} \\
&\Rightarrow \forall x \in Z, \exists y \in Y : x \in L^{Y,C}(y) \subseteq Y \ominus C \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by the definition of } L^{Y,C}(y)\text{)} \\
&\Rightarrow Z \subseteq Y \ominus C \qquad\qquad \text{(since } \forall x \in Z, x \in Y \ominus C\text{)} \\
&\Rightarrow Z \oplus C \subseteq (Y \ominus C) \oplus C \qquad\qquad \text{(by Proposition 2)} \\
&\Rightarrow Z \oplus C \subseteq (Y \ominus C) \oplus C = Y \qquad\qquad \text{(by Proposition 7)} \\
&\Rightarrow Z \oplus C \subseteq Y.
\end{aligned}
$$

Hence, we conclude that if Eq. (1) has any solution, then, for all $Z \in \Phi^{Y,C}$, $Z \oplus C = Y$. □

The next result shows that all lower bounds for the solutions are in $\Phi^{Y,C}$.

Theorem 2. *Let $C \in \mathcal{P}(E)$ and $Y \in \mathcal{P}(E)$. For all $X \in \mathcal{P}(E)$ such that $X \oplus C = Y$, there exists $Z \in \Phi^{Y,C}$, such that $Z \subseteq X$.*

Proof.

$$
\begin{aligned}
X \oplus C = Y &\Rightarrow \forall y \in Y, \exists x \in X, \exists c \in C, x + c = y \qquad\qquad \text{(by Proposition 8)} \\
&\Rightarrow \forall y \in Y, \exists x \in X, \exists c \in C, x = y - c \in Y \ominus C \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad (X \subseteq Y \ominus C, \text{ by Proposition 6}) \\
&\Rightarrow \forall y \in Y, \exists x \in X, x \in L^{Y,C}(y). \qquad \text{(by the definition of } L^{Y,C}(y)\text{)}
\end{aligned}
$$

Therefore, $\forall X \in \mathcal{P}(E)$ such that $X \oplus C = Y$, we have $\forall y \in Y, \exists x \in X$ such that $x \in L^{Y,C}(y)$. Hence, we denote $x_y \in X$ an element such that $x_y \in L^{Y,C}(y)$. Let us build $Z \in \mathcal{P}(E)$, as follows:

$$Z \in \mathcal{P}(E) : Z = \bigcup_{y \in Y} \{x_y\}.$$

By construction, we see that $Z \in \Phi^{Y,C}$. Furthermore, $Z \subseteq X$, since Z is built by elements in X. Then, for all $X \in \mathcal{P}(E)$ such that $X \oplus C = Y$, there exists $Z \in \Phi^{Y,C}$, such that $Z \subseteq X$. □

4.3 Set for All Solutions

Note that Proposition 7 states an upper bound for all solutions of the Minkowski equation, while Theorem 2 presents a set containing all the lower bounds. With this in mind, let us define a interval of sets. Let $A, B \in \mathcal{P}(E)$ such that $A \subseteq B$. We define an *interval* $[A, B] \subseteq \mathcal{P}(E)$ as a set given by $[A, B] = \{X \in \mathcal{P}(E) : A \subseteq X \subseteq B\}$. Thus, it is easy to see that if there exists $X \in \mathcal{P}(E)$ such that

$X \oplus C = Y$, then, there exists $Z \in \Phi^{Y,C}$ such that $X \in [Z, Y \ominus C]$. So, with this, we can build a *collection of intervals* that contains all solutions for the Minkowski equation, that is,

$$\mathbf{S}^{Y,C} = \{[Z, Y \ominus C] : Z \in \Phi^{Y,C}\}. \tag{4}$$

More surprisingly, all elements in the interval $[Z, Y \ominus C] \in \mathbf{S}^{Y,C}$ are solutions for the Minkowski equation.

Theorem 3. *Let $C \in \mathcal{P}(E)$ and $Y \in \mathcal{P}(E)$. If there exists $X \in \mathcal{P}(E)$ such that $X \oplus C = Y$, then, for all $Z \in \Phi^{Y,C}$ and for all $X \in [Z, Y \ominus C]$, $X \oplus C = Y$.*

Proof.

$$\begin{aligned}
X \in [Z, Y \ominus C] &\Rightarrow Z \subseteq X \subseteq Y \ominus C && \text{(by the definition of interval)}\\
&\Rightarrow Z \oplus C \subseteq X \oplus C \subseteq (Y \ominus C) \oplus C && \text{(by Proposition 2)}\\
&\Rightarrow Y \subseteq X \oplus C \subseteq (Y \ominus C) \oplus C && (Z \oplus C = Y, \text{ by Theorem 2})\\
&\Rightarrow Y \subseteq X \oplus C \subseteq Y && ((Y \ominus C) \oplus C = Y, \text{ by Proposition 7})\\
&\Rightarrow X \oplus C = Y. && \square
\end{aligned}$$

It is still possible to eliminate some intervals from the set $\mathbf{S}^{Y,C}$ without losing any solution. In fact, if there are two distinct sets Z_i and Z_j in $\Phi^{Y,C}$ such that $Z_i \subseteq Z_j$, then, obviously, $[Z_i, Y \ominus C] \supseteq [Z_j, Y \ominus C]$, and therefore, we can take $[Z_j, Y \ominus C]$ out from $\mathbf{S}^{Y,C}$. In fact, an interval $[A, B]$ is called *maximal* in a given collection of intervals $\mathbf{I}$ if and only if there does not exist an interval $[A', B']$ in $\mathbf{I}$, distinct of $[A, B]$, such that $[A', B'] \supseteq [A, B]$. We call a collection of intervals $\mathbf{I}$ as a *collection of maximal intervals* if and only if all its intervals are maximal in $\mathbf{I}$. So, the collection of intervals that contains all solutions for the Minkowski equation is a collection of maximal interval $\mathbf{X}^{Y,C}$ obtained directly from $\Phi^{Y,C}$:

$$\mathbf{X}^{Y,C} = \{[Z, Y \ominus C] : Z \in \Phi^{Y,C}\text{and } \nexists Z' \in \Phi^{Y,C}, Z' \neq Z, Z' \subseteq Z\}. \tag{5}$$

Let us give an example for the construction of $\mathbf{X}^{Y,C}$.

Example 3. For this example, consider the sets described in Example 2. Observe that the subset $\mathtt{10\underline{1}01} \in \Phi^{Y,C}$ is contained in subsets $\mathtt{11\underline{1}11}$, $\mathtt{10\underline{1}11}$, $\mathtt{11\underline{1}01} \in \Phi^{Y,C}$ (and therefore, these subsets are not considered in the final solution set), and there is no subset in $\Phi^{Y,C}$ that is contained in $\mathtt{11\underline{0}11}$. Then, the final solution is $\mathbf{X}^{Y,C} = \{[\mathtt{10\underline{1}01}, \mathtt{11\underline{1}11}],\ [\mathtt{11\underline{0}11}, \mathtt{11\underline{1}11}]\}$. □

An important feature of the analytical solution formula is that all solutions of the Minkowski equation are given in a compact representation (a collection of maximal intervals), that is, the number of intervals in $\mathbf{X}^{Y,C}$ is smaller than, or in the worst case equal to, the number of all possible solutions.

Let us present an algorithm (see Algorithm 1) based on the results of this section that, given a subset $Y \in \mathcal{P}(E)$ and a subset $C \in \mathcal{P}(E)$, computes $\mathbf{X}^{Y,C}$.

```
Algorithm 1: FindSolMinkEq ($Y, C$)
    Input: A subset $Y \in \mathcal{P}(E)$ and a subset $C \in \mathcal{P}(E)$.
    Output: A collection of maximal intervals $\mathbf{X}^{Y,C}$ that contains all solutions of
            the equation $X \oplus C = Y$.
/* Verify the existence of solutions (based on Proposition 7) */
if $Y \ominus C \oplus C \neq Y$ then
    return "no solution exists";
end
/* Compute the sets $L^{Y,C}(y)$ (see Eq. 1)*/
forall the $y \in Y$ do
    $L^{Y,C}(y) = \{x \in Y \ominus C : x = y - c, c \in C\}$;
end
/* Compute the set $\Phi^{Y,C}$ (see Eq. 3) */
$\Phi^{Y,C} = \{Z \in \mathcal{P}(E) : Z = \bigcup_{i=1}^{|Y|} \{x_i\}, (x_1, x_2, \ldots, x_{|Y|}) \in \prod_{y \in Y} L^{Y,C}(y)\}$;
/* Compute $\mathbf{X}^{Y,C}$, a collection of maximal intervals (see Eq. 5) */
$\mathbf{X}^{Y,C} = \{[Z, Y \ominus C] : Z \in \Phi^{Y,C}$and $\nexists Z' \in \Phi^{Y,C}, Z' \neq Z, Z' \subseteq Z\}$;
return $\mathbf{X}^{Y,C}$;
```

The most time-consuming step of Algorithm 1 is the construction of $\Phi^{Y,C}$ which is $O(|C|^{|Y|})$. As we can see, this step makes Algorithm 1 an exponential time complexity algorithm (the other steps can be executed in polynomial time). But, it is not that surprising, since the algorithm outputs all solutions of the Minkowski equation, which in turn can be, in worst case, exponential.

4.4 Structuring Element Decomposition

A direct application of Algorithm 1 is to find a decomposition of a structuring element in terms of Minkowski addition of smaller subsets. More formally, let Y be a finite subset of E and let $\mathcal{Q} = \{B_1, B_2, \ldots, B_k\}$ a collection of structuring elements (for example, the powerset of Y or the set of all subsets of the elementary square – the 3×3 square centered at the origin). The problem is to find a sequence $[Z_1, Z_2, \ldots, Z_k]$ of elements of $\mathcal{Q}$ which is a sequential decomposition of Y, that is, $Y = Z_1 \oplus \cdots \oplus Z_k$, where each $Z_i \in \mathcal{Q}$. We call this decomposition a *sequential decomposition of* Y *in terms of* $\mathcal{Q}$. It is possible to apply recursively the algorithm that find all solutions of the Minkowski equation (Algorithm 1) in an appropriate way to obtain such a decomposition (see Algorithm 2).

We have two observations for Algorithm 2. The first one is that its time complexity is clearly exponential and thereby Algorithm 2 is not plausible for using in real situations. For achieving an efficient algorithm, more research must be done. The second observation is that, if we place the structuring elements of $\mathcal{Q}$ in increasing order and they are visited in this order at Line 4, then the decomposition found by Algorithm 2 (if one exists) has the smallest total number of points.

Algorithm 2: FindSeqDecomp $(Y, \mathcal{Q})$

```
Input: A subset $Y \in \mathcal{P}(E)$ and a collection of structuring elements
       $\mathcal{Q} = \{B_1, B_2, \ldots, B_k\}$.
Output: A sequence $[Z_1, Z_2, \ldots, Z_k]$ which is sequential decomposition of $Y$ in
        terms of elements of $\mathcal{Q}$ (if one exists), or an empty sequence $[\,]$ if $Y$ is
        not decomposable.
if $Y \in \mathcal{Q}$ then
    return $[Y]$;
end
foreach $B \in \mathcal{Q}$ do
    /* Let $\mathbf{X}^{Y,B}$ be the output of the Algorithm 1 */
    $\mathbf{X}^{Y,B} \leftarrow$ FindSolMinkEq $(Y, B)$;
    foreach $[Z, Y \ominus B] \in \mathbf{X}^{Y,B}$ do
        foreach $X \in [Z, Y \ominus B]$ do
            $[Z_1, Z_2, \ldots, Z_k] \leftarrow$ FindSeqDecomp $(X, \mathcal{Q})$;
            if $[Z_1, Z_2, \ldots, Z_k] \neq [\,]$ then
                return $[Z_1, Z_2, \ldots, Z_k, B]$;
            end
        end
    end
end
/* Empty Sequence */
return $[\,]$;
```

5 Conclusion

In this paper, we presented the formulation of a discrete equation (called *Minkowski Addition Equation*) whose solution space has a strong combinatory nature. The main (original) result presented in this paper (which is a theoretical result) is an analytical solution formula for this equation.

This analytical formula is obtained by using a set of lower bounds and an upper bound for the solutions of the equation. The final result is a (compact representation) formula which is expressed as a collection of maximal intervals. A natural continuation of this work is to study the extension of this result for functions (gray-level images).

One direct application of the solutions of this equation is that they can be used to find best representations for fast computation of erosions and dilations in terms of a sequential decomposition its structuring elements. However, an algorithm for finding such decomposition using the analytical formula has exponential time complexity. In this way, for future research, more study must be done in order to obtain faster algorithms (for example, using heuristic techniques) for structuring element decomposition.

Acknowledgments. The authors are grateful to FAPESP and CNPq for the financial support.

References

1. Rådström, H.: An embedding theorem for spaces of convex sets. Proc. Amer. Math. Soc. 3, 165–169 (1952)
2. Hörmander, L.: Sur la fonction d'appui des ensembles convexes dans un espace localement convexe. Arkiv för Matematik 3(2), 181–186 (1955)
3. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, New York (1982)
4. Maragos, P.A.: A Unified Theory of Translation-invariant Systems with Applications to Morphological Analysis and Coding of Images. PhD thesis, School of Elect. Eng. - Georgia Inst. Tech. (1985)
5. Hashimoto, R.F., Barrera, J., Ferreira, C.E.: A Combinatorial Optimization Technique for the Sequential Decomposition of Erosions and Dilations. Journal of Mathematical Imaging and Vision 13(1), 17–33 (2000)
6. Hashimoto, R.F., Barrera, J.: A Note on Park and Chin's Algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 139–144 (2002)
7. Hashimoto, R.F., Barrera, J.: A Greedy Algorithm for Decomposing Convex Structuring Elements. Journal of Mathematical Imaging and Vision 18(3), 269–289 (2003)
8. Park, H., Chin, R.T.: Decomposition of Arbitrarily Shaped Morphological Structuring Elements. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(1), 2–15 (1995)
9. Shih, F.Y., Wu, Y.T.: Decomposition of binary morphological structuring elements based on genetic algorithms. Computer Vision and Image Understanding 99(2), 291–302 (2005)
10. Zhang, C.B., Wang, K.F., Shen, X.L.: Algorithm for decomposition of binary morphological convex structuring elements based on periodic lines. Imaging Science Journal (May 2012)
11. Zhuang, X., Haralick, R.: Morphological Structuring Element Decomposition. Computer Vision, Graphics, and Image Processing 35, 370–382 (1986)
12. Babu, G., Srikrishna, A., Challa, K., Babu, B.: An error free compression algorithm using morphological decomposition. In: 2012 International Conference on Recent Advances in Computing and Software Systems (RACSS), pp. 33–36 (April 2012)
13. Heijmans, H.J.A.M.: Morphological Image Operators. Academic Press, Boston (1994)

A Comparison of Many Max-tree Computation Algorithms

Edwin Carlinet[*] and Thierry Géraud

EPITA Research and Development Laboratory (LRDE)
{edwin.carlinet,thierry.geraud}@lrde.epita.fr

Abstract. With the development of connected filters in the last decade, many algorithms have been proposed to compute the max-tree. Max-tree allows computation of the most advanced connected operators in a simple way. However, no exhaustive comparison of these algorithms has been proposed so far and the choice of an algorithm over another depends on many parameters. Since the need for fast algorithms is obvious for production code, we present an in depth comparison of five algorithms and some variations of them in a unique framework. Finally, a decision tree will be proposed to help the user choose the most appropriate algorithm according to their requirements.

1 Introduction

In mathematical morphology, connected filters are those that modify an original signal by only removing connected components, hence those that preserve image contours. Originally, they were mostly used for image filtering [19, 16]. Major advances came from max and min-tree as hierarchical representations of connected components and from an efficient algorithm able to compute them [15]. Since then, usage of these trees has soared for more advanced forms of filtering: based on attributes [4], using new filtering strategies [15, 18], allowing new types of connectivity [12]. They are also a base for other image representations. In [9] a tree of shapes is computed from a merge of min and max trees. In [24] a component tree is computed over the attributes values of the max-tree. Max-trees have been used in many applications: computer vision through motion extraction [15], features extraction with MSER [6], segmentation, 3D visualization [20]. With the increase of applications comes an increase of data type to process: 12-bit images in medical imagery [20], 16-bit or float images in astronomical imagery [1], and even multivariate data with special ordering relation [13]. With the improvement of optical sensors, images are getting bigger (so do image data sets) which argues for the need for fast algorithms. Many algorithms have been proposed to compute the max-tree efficiently but only partial comparisons have been proposed. Moreover, some of them are dedicated to a particular task (e.g., filtering) and are unusable for other purposes. In this paper, we provide a full and exhaustive comparison of state-of-the-art max-tree algorithms in a unique framework, i.e.,

[*] Edwin Carlinet is now also with Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Équipe A3SI, ESIEE Paris, France.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 73–85, 2013.

same architecture, same language (C++) and same outputs. The paper is organized as follows: Section 2 recalls basic notions and manipulations of max-tree, describes algorithms and implementations used in this study. Section 3 is dedicated to the comparison of those algorithms both in terms of complexity and running times through experimentations.

2 A Tour of Max-tree: Definition, Representation and Algorithms

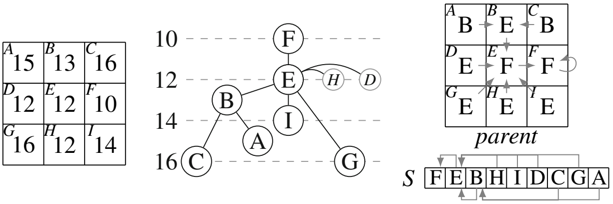
2.1 Basic Notions for Max-tree

Let $ima : \Omega \rightarrow V$ an image on regular domain Ω, having values on a totally preordered set $(V, <)$ and let $\mathcal{N}$ a neighborhood on Ω. Let $\lambda \in V$, we note $[ima \leq \lambda]$ the set $\{p \in \Omega, ima(p) \leq \lambda\}$. Let $X \subset \Omega$, we note $CC(X) \subset \mathcal{P}(\Omega)$ the set of connected components of X w.r.t the neighborhood $\mathcal{N}$; $\mathcal{P}(\Omega)$ being the power set of all the possible subsets of Ω. $\{CC([ima = \lambda]), \lambda \in V\}$ are *level components* and $\{CC([ima \geq \lambda]), \lambda \in V\}$ (resp. $\leq$) is the set of upper components (resp. lower components). The latter endowed with the inclusion relation form a tree called the max-tree (resp. min-tree). Since min and max-trees are dual, this study obviously holds for min-tree as well. Finally, the peak component of p at level λ noted P_p^λ is the upper component $X \in CC([ima \geq \lambda])$ such that $p \in X$.

2.2 Max-tree Representation

Berger et al. [1], Najman and Couprie [10] rely on a simple and effective encoding of component-trees using an image that stores the *parent* relationship that exists between components. A connected component is represented by a single point called the *canonical element* [1, 10] or *level root*. Let two points $p, q \in \Omega$, and p_r the root of the tree. We say that p is canonical if $p = p_r$ or $ima(parent(p)) < ima(p)$. A *parent* image shall verify the following three properties: 1) $parent(p) = p \Rightarrow p = p_r$ - the root points to itself and it is the only point verifying this property - 2) $ima(parent(p)) \leq ima(p)$ and 3) $parent(p)$ is canonical.

Furthermore, having just the *parent* image is an incomplete representation since it is not sufficient to easily perform classical tree traversals. For that, we need an extra array of points, $S : \mathbb{N} \rightarrow \Omega$, where points are stored so that



(a) Original image *ima* (b) Max-tree of *ima* (c) Max-tree representation

Fig. 1. Representation of a max-tree with a parent image and an array

$\forall i, j \in \mathbb{N}\ i < j \Rightarrow S[j] \neq parent(S[i])$. Thus browsing S elements allows to traverse the tree downwards, whereas a reverse browsing of S is an upward tree traversal. Note that having both S and *parent* thus makes it useless to store the children of each node. Figure 1 shows an example of such a representation of a max-tree. This representation only requires $2nI$ bytes memory space where n is the number of pixels and I the size in bytes of an integer, since points stored in S and *parent* are actually positive offsets in a pixel buffer. The algorithms we compare have all been modified to output the same tree encoding, that is, the couple $(parent,\ S)$.

2.3 Attribute Filtering and Reconstruction

A classical approach for object detection and filtering is to compute some features called attributes on max-tree nodes. A usual attribute is the number of pixels in components. Followed by a filtering, it leads to the well-known area opening. More advanced attributes have been used like elongation, moment of inertia [22] or even mumford-shah like energy [24]. Some max-tree algorithms [21, 6] construct the *parent* image only; they do not compute S. As a consequence, they do not provide a "versatile" tree, i.e., a tree that can be easily traversed upwards and downwards, that allows attribute computation and non-trivial filtering. Here we require every algorithms to output a "complete" tree representation (*parent and S*) so that it can be multi-purposedly usable. The rationale behind this requirement is that, for some applications, filtering parameters are not known yet at the time the tree is built (e.g., for interactive visualization [20]). In the algorithms we compare in this paper, no attribute computation nor filtering are performed during tree construction for clarity reasons; yet they can be augmented to compute attribute and filtering at the same time. Algorithm 1 provides an implementation of attribute computation and direct-filtering with the representation. $\hat{f} : \Omega \times V \to \mathcal{A}$ is an application that projects a pixel p and its value $ima(p)$ in the attribute space $\mathcal{A}$. $\hat{+} : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ is an associative operator used to merge attributes of different nodes. COMPUTE-ATTRIBUTE starts with computing attributes of each singleton node and merges them from leaves toward root. Note that this simple code relies on the fact that a node receives all information from its children before passing its attribute to the parent. Without any ordering on S, it would not have been possible. DIRECT-FILTER is an implementation of direct filtering as explained in [15] that keeps all nodes passing a criterion λ and lowers nodes that fails to the last ancestor "alive". This implementation has to be compared with the one in [21] that only uses *parent*. This one is shorter, faster and clearer above all.

2.4 Three Kinds of Max-tree Algorithms

Max-tree algorithms can be classified in three classes.

Immersion Algorithms. They start with building n disjoints singleton for each pixel and sort them according to their gray value. When immersion starts, highest levels are processed first such that local maxima create some bassins. While processing pixels in decreasing order of their gray value, bassins that form disjoint sets of pixels are extended and merges when the current pixel creates a connection

Algorithm 1. Computation of attributes and filtering

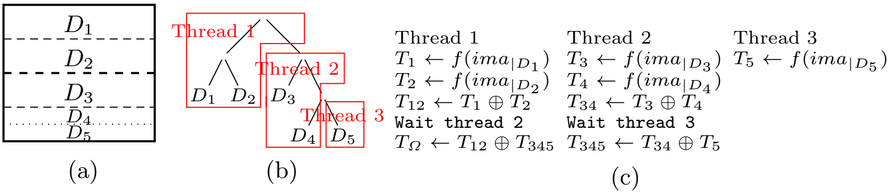
```
function
COMPUTE-ATTRIBUTE($S$, $parent$, $ima$)
    $p_{root} \leftarrow S[0]$
    for all $p \in S$ do
        $attr(p) \leftarrow \hat{f}(p, ima(p))$
    for all $p \in S$ backward, $p \neq p_{root}$ do
        $q \leftarrow parent(p)$
        $attr(q) \leftarrow attr(q) \hat{+} attr(p)$
    return $attr$
```

```
function
DIRECT-FILTER($S$, $parent$, $ima$, $attr$)
    $p_{root} \leftarrow S[0]$
    if $attr(p_{root}) < \lambda$ then $out(p_{root}) \leftarrow 0$
    else $out(p_{root}) \leftarrow ima(p_{root})$
    for all $p \in S$ forward do
        $q \leftarrow parent(p)$
        if $ima(q) = ima(p)$ then
            $out(p) \leftarrow out(q)$   ▷ $p$ not canonical
        else if $attr(p) < \lambda$ then
            $out(p) \leftarrow out(q)$   ▷ Criterion failed
        else
            $out(p) \leftarrow ima(p)$   ▷ Criterion pass
    return $out$
```

between two of them. Finally, the pixels at lowest level are processed, all bassins have been merged and the whole image has been immersed. The way bassins grow and merge form a tree. Disjoint connected sets of pixels are handled with Tarjan's union-find algorithm where connected sets are encoded with trees. A single pixel, the root of the tree, represents the whole connected component. Tarjan [17] provides three basic manipulation routines: MAKE-SET(p) to create the singleton set $\{p\}$, FIND-ROOT(p) to get the root of the component that contains p, and MERGE-SET(p, q) that merge two disjoints sets of roots p and q. Tarjan discussed two important optimizations in [17] for union-find: root path compression and union-by-rank. Root path compression takes part in FIND-ROOT(p), points on the path from p to the root collapse to the actual root the component. Union-by-rank takes place in MERGE-SET(p, q), when merging two components rooted in p and q, we have to select one to represent the newly created component. If the component of p has a *rank* greater than the one of q then p is selected as the new root, q otherwise. When rank matches the depth of trees, it enables tree balancing and guaranties a better complexity for union-find. Path compression has been applied in [1] and [10], while union-by-rank only in [10].

Flooding Algorithms. Those start from a flooding point and perform a propagation. Points in the propagation front are stored in a priority queue so that points at highest level are flooded first, i.e., a depth first propagation. A first implementation has been proposed by [15] which relies on a recursive function FLOOD(p) in charge of flooding p at level $\lambda = ima(p)$ and all points in P_p^λ. When FLOOD(p) returns, the corresponding node has been constructed and is attached to its parent. Hence, when FLOOD(p_{min}) terminates, where p_{min} is a point a lowest gray level in ima, the whole image has been flooded and the tree is constructed. To speedup the algorithm, the propagation priority queue is encoded with a hierarchical queue that offers constant time PUSH and POP operations and direct access to points at any level. Salembier et al. [15]'s algorithm was rewritten in a non-recursive implementation by Hesselink [3] and later by Nistér and Stewénius [11] and Wilkinson [23]. These algorithms differ in only two points. First, [23] uses a pass to retrieve the root before flooding to mimic the original recursive version while Nistér and Stewénius [11] does not. Second, priority queues in [11] use an unacknowledged implementation of heap based on hierarchical queues while in [23] they are implemented using a standard heap (based on comparisons).

Merge-Based Algorithms. Merge-based algorithms consist in computing max-tree on sub-parts of images and merging back trees to get the max-tree of the whole image. Those algorithms are typically well-suited for parallelism since they adopt a map-reduce idiom [21]. Computation of sub max-trees (map step), done by any sequential method, and merge (reduce-step) are executed in parallel by several threads. In order to improve cache coherence, images should be split in contiguous memory blocks that is, splitting along the first dimension if images are row-major. When blocks are image lines, a dedicated 1D max-tree algorithm can be used [7, 8]. Figure 2 shows an example of parallel processing using map-reduce idiom. Choosing the right number of splits and jobs distribution between threads is a difficult topic that depends on the architecture (number of threads available, power frequency of each core). If the domain is not split enough (number of chunks no greater than number of threads) the parallelism is not maximal, some threads become idle once they have done their jobs, or wait for other thread to merge. On the other hand, if the number of split gets too large, merging and thread synchronization cause significant overheads. Since work balancing and thread management are outside the current topic, they are delegated to high level parallelism library such as Intel's TBB [14].



Thread 1	Thread 2	Thread 3
$T_1 \leftarrow f(ima_{\vert D_1})$	$T_3 \leftarrow f(ima_{\vert D_3})$	$T_5 \leftarrow f(ima_{\vert D_5})$
$T_2 \leftarrow f(ima_{\vert D_2})$	$T_4 \leftarrow f(ima_{\vert D_4})$	
$T_{12} \leftarrow T_1 \oplus T_2$	$T_{34} \leftarrow T_3 \oplus T_4$	
`Wait thread 2`	`Wait thread 3`	
$T_\Omega \leftarrow T_{12} \oplus T_{345}$	$T_{345} \leftarrow T_{34} \oplus T_5$	

(a) (b) (c)

Fig. 2. Map-reduce idiom for max-tree computation. (a) Sub-domains of *ima*. (b) A possible distribution of jobs by threads. (c) Map-reduce operations where f is the map operator, $\oplus$ the merge operator, and $ima_{|D_n}$ denotes the image restricted to sub-domain D_n.

2.5 Algorithms Implementation

Immersion Algorithms. An implementation of max-tree based on union-find without union-by-rank can be found in [1] and an implementation using union-by-rank in [10]. We adapted the latter to build *parent* and S without extra cost. Here, we only provide the principles of our new implementation of union-by-rank and a lighter that uses less memory called level compression.

Union-by-Rank. Our implementation is similar to that of [1] but augmented with union-by-rank. The basis of the algorithm resides in two images *parent* and *zpar* representing two trees. *Parent* encodes the max-tree while *zpar* the underlying tree for tracking disjoint sets with union-find. Without *rank* balancing, root of any component in *zpar* matches the root of the corresponding max-tree in *parent*. When using union-by-rank to merge components from *zpar*, we loose this property. Therefore, we introduce an new image *repr* that keeps a connection between the root of the set in *zpar* and the root of the max-tree in *parent* updated. This method is slightly different from the one of Najman and Couprie [10]. They use two union-find structures, one of them dedicated to handle flat zones, while our implementation only uses a single one for *zpar*.

Level Compression. Union-by-rank provides time complexity guaranties at the price of extra memory requirement. When dealing with huge images this results in a significant drawback (e.g. RAM overflow...) Without rank technique, the last point processed always becomes the root of the component, i.e. in most cases, we merge a deep tree to a single node that tends to create a degenerated tree in flat zones. Level compression avoids this behavior by a special handling of flat zones. Let p be the point in process at level λ, n a neighbor of p already processed, z_p the root of P_p^λ (at first $z_p = p$), z_n the root of P_n^λ. If z_p and z_n have a same gray level, they belong to the same node and we can choose any of them as a canonical element. Normally z_p should become the root with child z_n but level compression inverts the relation: z_n is kept as the root and z_p becomes a child. The remaining part of the algorithm stays unchanged *w.r.t.* [1].

Flooding Algorithms. The first recursive flooding algorithm was proposed in [15]. A non-recursive version implemented with hierarchical queues can be found in [11], and the one relying on a standard heap in [23]. Those algorithms have been slightly modified to use *parent* and S representation with no other modification.

Merge-Based Algorithms. Algorithms used to merge two trees can be found in [21, 7]. As, implementation of the special 1D max-tree algorithm used when sub-domains are image lines has been proposed in [8]. Our implementations match the ones proposed in those papers. The major difference resides in a post-processing to ensure tree canonicalization and S construction. Indeed, once sub-trees have been computed and merged into a single tree, it does not hold any canonical property (because non-canonical elements are not updated during merge). In addition, the reduction step does not merge the S array corresponding to sub-trees (it would imply reordering S which is more costly than just recomputing it at the end). Algorithm 2 performs canonicalization and reconstructs the S array from *parent* image. It uses an auxiliary image *dejavu* to track nodes that have already been inserted in S. As opposed to other max-tree algorithms, construction of S and processing of nodes are top-down. For any point p, we traverse in a recursive way its path to the root to process its ancestors. When the recursive call returns, *parent*(p) is already inserted into S and holds the canonical property, thus we can safely insert p back in S and canonicalize p.

Implementation Details. Algorithms have been implemented in pure C++ using an STL implementation of some basic data structures (heaps, priority queues), Milena[5] image processing library to provide fundamental image types and I/O functionality, and Intel TBB for parallelism. Specific implementation

Algorithm 2. Canonicalization and S computation algorithm

```
procedure CanonizeRec(p)
    $dejavu(p) \leftarrow true$
    $q \leftarrow parent(p)$
    if not $dejavu(q)$ then                              ▷ Process parent before $p$
        CanonizeRec(q)
    if $ima(q) = ima(parent(q))$ then                    ▷ Canonize
        $parent(p) \leftarrow parent(q)$
    InsertBack($S, p$)

for all $p$ do $dejavu(p) \leftarrow False$
for all $p \in \Omega$ such that not $dejavu(p)$ do
    CanonizeRec($p$)
```

optimizations are the following. Sorting is optimized by switching to counting sort when quantization is lower than 18 bits. For large integers of q bits, it switches to 2^{16}-based radix sort requiring $q/16$ counting sort. For immersion algorithms, queues and stacks are *pre-allocated* to avoid dynamic memory reallocation. Hierarchical queues are also *pre-allocated* by computing image histogram as a pre-processing. In our non-recursive implementation of Salembier (called non-recursive Salembier below) priority-queues are implemented with hierarchical queues (i.e., Nistér and Stewénius [11]'s implementation) and switches to the STL standard heap implementation[1], with a *pre-allocation* for data as well, when the number of bits exceeds 18 (i.e., Wilkinson [23]'s one). A y-fast trie can be used for large integers ensuring a better complexity (see Section 3.1) but no performance gain has been obtained. Finally, in parallel versions of the algorithms, all instructions that deal about S construction and *parent* canonicalization have been removed since S is reconstructed and *parent* canonicalized by Algorithm 2.

3 Algorithms Comparison

3.1 Complexity Analysis

Let $n = H * W$ with H the image height, W the image width and n the total number of pixels. Let k be the number of values in V.

Immersion Algorithms require sorting pixels, a process of $\Theta(n + k)$ complexity ($k \ll n$) for small integers (counting sort), $O(n \log \log n)$ for large integers (hybrid radix sort), and $O(n \log n)$ for generic data types with a more complicated ordering relation (comparison sort). Union-find is $O(n \log n)$ and $O(n\alpha(n))$ when used with union-by-rank. [2]. Canonicalization is linear and does not use extra memory. Memory-wise, sorting may require an auxiliary buffer depending on the algorithm and histograms for integer sorts thus $\Theta(n + k)$ extra-space. Union without rank requires a *zpar* image for path compression ($\Theta(n)$) and the system

[1] Please note that the authors have verified that choosing a particular implementation of STL (namely STLport 4.6 vs. gcc 4.7) does *not* impact the results presented here.

[2] $\alpha(n)$, the inverse of Ackermann function, is very low growing, $\alpha(10^{80}) \simeq 4$.

stack for recursive calls in `findroot` which is $O(n)$ (`findroot` could be non-recursive, but memory space is saved at cost of a higher computational time). Union-by-rank requires two extra images (*rank* and *repr*) of n pixels each.

Flooding Algorithms require a priority queue to retrieve the highest point in the propagation front. Each point is inserted and removed once, thus the complexity is $\Theta(np)$ where p is the cost of pushing or popping from the heap. If the priority queue is encoded with a hierarchical queue as in [15] or [11], it uses $n + 2k$ memory space, provides constant insertion and constant access to the maximum but popping is $O(k)$. In practice, in images with small integers, gray level difference between neighboring pixels is far to be as large as k. With high dynamic image, a heap can be implemented with a y-fast trie, which has insertion and deletion in $O(\log \log k)$ and access to maximum element in $O(1)$. For any other data type, a "standard" heap based on comparisons requires n extra space, allows insertion and deletion in $O(\log n)$ and has a constant access to its maximal element. Those algorithms need an array or a stack of respective size k and n. Salembier's algorithm uses the system stack for a recursion of maximum depth k, hence $O(k)$ extra-space.

Merge-Based Algorithms complexity depends on $\mathcal{A}(k, n)$, the complexity of the underlying method used to compute max-tree on sub-domains. Let $s = 2^h$ the number of sub-domains. The map-reduce algorithms requires s mapping operations and $s - 1$ merges. A good map-reduce algorithm would split the domain to form a full and complete tree so we assume all leaves to be at level h. Merging sub-trees of size $n/2$ has been analyzed in [21] and is $O(k \log n)$ (we merge nodes of every k levels using union-find without union-by-rank). Thus, the complexity of a single reduction is $O(Wk \log n)$. Assuming s constant and $H = W = \sqrt{n}$ the complexity as a function of n and k of the map-reduce algorithm is $O(\mathcal{A}(k, n)) + O(k\sqrt{n} \log n)$. When there is as many splits as rows, s in now dependent on n. This leads to the Matas et al. [7] algorithm whose complexity is $O(n) + O(k\sqrt{n}(\log n)^2)$. Contrary to what they claim, when values are small integers the complexity stays linear and is not dominated by merging operations. Finally, canonicalization and S reconstruction have a linear time complexity (`CanonizeRec` is called only once for each point) and only use an image of n elements to track already processed points.

3.2 Experiments

Benchmarks were performed on an Intel Core i7 (4 physical cores, 8 logical cores). The programs were compiled with gcc 4.7, optimization flags on (`-O3 -march=native`). Tests were conducted on a 6.8 MB 8-bit image which was re-sized by cropping or tiling the original image. Over-quantization was performed by shifting the eight bits left and generating missing lower bits at random. Figure 3 depicts performance of sequential algorithms w.r.t to image size and quantization. As a first remark, we notice that all algorithms are linear in practice. On natural images, the upper bound $n \log n$ complexity of the Wilkinson [23] and Berger et al. [1] algorithms is not reached. Let start with union-find

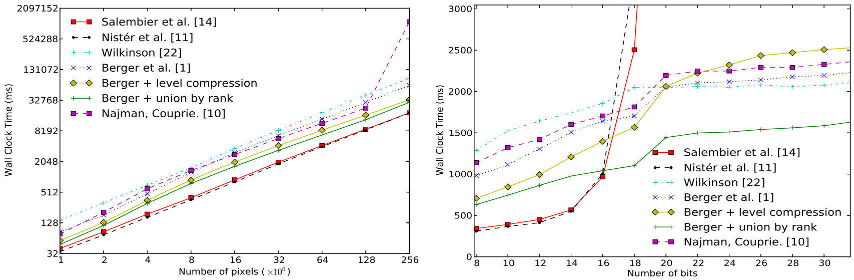
[3] Note always available because hierarchical queues requires V to be expressible as an index.

Table 1. Time complexity of many max-tree algorithms compared. n is the number of pixels and k the number of gray levels.

	Time Complexity		
Algorithm	Small int	Large int	Generic V
Berger [1]	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Berger + rank	$O(n\,\alpha(n))$	$O(n \log\log n)$	$O(n \log n)$
Najman and Couprie [10]	$O(n\,\alpha(n))$	$O(n \log\log n)$	$O(n \log n)$
Salembier et al. [15]	$O(nk)$	$O(nk) \simeq O(n^2)$	N/A [3]
Nistér and Stewénius [11]	$O(nk)$	$O(nk) \simeq O(n^2)$	N/A[3]
Wilkinson [23]	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Salembier non-recursive	$O(nk)$	$O(n \log\log n)$	$O(n \log n)$
Menotti et al. [8] (1D)	$O(n)$	$O(n)$	$O(n)$
Map-reduce	$O(A(k,n))$	$O(A(k,n))+$ $O(k\sqrt{n}\log n)$	$O(A(k,n))+$ $O(k\sqrt{n}\log n)$
Matas et al. [7]	$O(n)$	$O(n) + O(k\sqrt{n}(\log n)^2)$	-

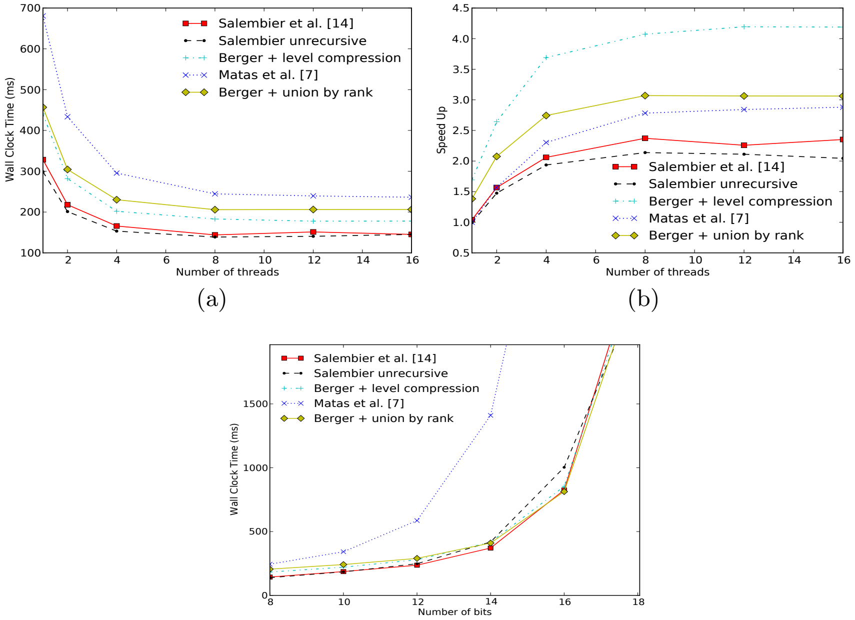
Table 2. Space requirements of many max-tree algorithms compared. n is the number of pixels and k the number of gray levels.

	Auxiliary space requirements		
Algorithm	Small int	Large int	Generic V
Berger et al. [1]	$n + k + stack$	$2n + stack$	$n + stack$
Berger + rank	$3n + k + stack$	$4n + stack$	$3n + stack$
Najman and Couprie [10]	$5n + k + stack$	$6n + stack$	$5n + stack$
Salembier et al. [15]	$3k + n + stack$	$2k + n + stack$	N/A[3]
Nistér and Stewénius [11]	$2k + 2n$	$2k + 2n$	N/A[3]
Wilkinson [23]	$3n$	$3n$	$3n$
Salembier non-recursive	$2k + 2n$	$3n$	$3n$
Menotti et al. [8] (1D)	k	n	n
Matas et al. [7]	$k + n$	$2n$	$2n$
Map-reduce	$\ldots + n$	$\ldots + n$	$\ldots + n$



(a) (b)

Fig. 3. (a) Algorithms comparison on a 8-bit image as a function of size; (b) Algorithms comparison on a 6.8 Mega-pixels image as a function of quantization.



(a)

(b)

(c)

Fig. 4. (a,b) Parallel algorithms comparison on a 6.8 Mega-pixels 8-bits image as a function of number of threads. (a) Wall clock time; (b) speedup w.r.t the sequential version; (c) Parallel algorithms comparison using 8 threads on a 6.8 Mega-pixels image as a function of quantization.

based algorithms. Berger et al. [1] and Najman and Couprie [10] have quite the same running time (±6% on average), however the performance of Najman and Couprie [10] algorithm drops significantly at 256 Mega-pixels. Indeed, at that size each auxiliary array/image requires 1 GB memory space, thus Najman and Couprie [10] who use a lot of memory exceed the 6 GB RAM limit and need to swap. Our implementation of union-by-rank uses less memory and is on average 42% faster than Najman and Couprie [10]. Level compression is an efficient optimization that provides 35% speedup on average on Berger et al. [1]. However, this optimization is only reliable on low quantized data. Figure 3b shows that it is relevant up to 18 bits. It operates on flat-zones but when quantization gets higher, flat-zones are less probable and the tests add worthless overheads. Union-find is not affected by the quantization but sorting does, counting sort and radix sort complexities are respectively linear and logarithmic with the number of bits. The break in union-find curves between 18 and 20 bits stands for the switch from counting to radix sort. Flooding-based algorithms using hierarchical queues outperform our union-find by rank on low quantized image by 41% on average. As expected, Salembier et al. [15] and Nistér and Stewénius [11] (which is the exact non-recursive version of the former) closely match. However, the exponential cost of hierarchical queues w.r.t the number of bits is evident on Figure 3b. By using a standard heap instead of hierarchical queues, Wilkinson

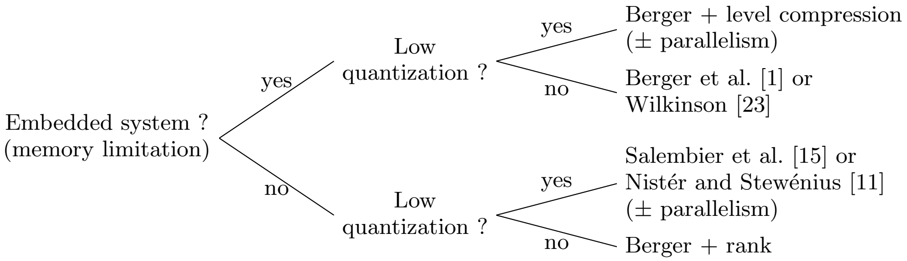


Fig. 5. Decision tree to choose the appropriate max-tree algorithm

[23] does scale well with the number of bits and outperforms every algorithms except our implementation of union-by-rank. In [23], the algorithm is supposed to match Salembier et al. [15]'s method for low quantized images, but in our experiments it remains 4 times slower. Since Najman and Couprie [10]'s algorithm is always outperformed by our implementation of union-find by rank, it will not be tested any further. Furthermore, because of the strong similarities of [11] and [23], they are merged in our single implementation (called *Non-recursive Salembier* below) that will use hierarchical queues when quantization is below 18 bits and switches to a standard heap implementation otherwise. Finally, the algorithm *Berger + level compression* will enable level compression only when the number of bits is below 18.

Figure 4 shows the results of the map-reduce idiom applied on many algorithms and their parallel versions. As a first result, we can see that better performance is generally achieved with 8 threads that is when the number of threads matches the number of (logical) cores. However, since there are actually only 4 physical cores, we can expect a ×4 maximum speedup. Some algorithms benefit more from map-reduce than others. Union-find based algorithms are particularly well-suited for parallelism. Union-find with level compression achieves the best speedup (×4.2) at 12 threads and union-find by rank a ×3.1 speedup with 8 threads. More surprising, the map-reduce pattern achieves significant speedup even when a single thread is used (×1.7 and ×1.4 for union-find with level compression and union-find by rank respectively). This result is explained by a better cache coherence when working on sub-domains that balance tree merges overheads. On the other hand, flooding algorithms do not scale as well because they are limited by canonicalization and S reconstruction post-process (that is going to happen as well for union-find algorithms on architectures with more cores). In [21] and [7], they obtain a speedup almost linear with the number of threads because only a *parent* image is built. If we remove the canonicalization and the S construction steps, we also get those speedups. Figure 4c shows the exponential complexity of merging trees as number of bits increases that makes parallel algorithms unsuitable for high quantized data. In light of the previous analysis, Figure 5 provides some guidelines on how to choose the appropriate max-tree algorithm *w.r.t.* to image types and architectures.

4 Conclusion

In this paper, we tried to lead a fair comparison of max-tree algorithms in a unique framework. We highlighted the fact that there is no such thing as the "best" algorithm that outranks all the others in every case and we provided a decision tree to choose the appropriate algorithm *w.r.t.* to data and hardware. We proposed a max-tree algorithm using union-by-rank that outperforms the existing one from [10]. Furthermore, we proposed a second one that uses level compression for systems with strict memory constraints. As further work, we shall include image contents as a new parameter of comparison, for instance images with large flat zones (e.g. cartoons) or images having strongly non-uniform distribution of gray levels. Extra-materials including algorithm pseudo-codes and descriptions can be found in the appendix of this paper [2] and a "reproducible research" code, intensively tested, is available on the Internet at `http://www.lrde.epita.fr/Olena/maxtree`.

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: Proc. of ICIP, vol. 41, pp. IV–41 (2007)

[2] Carlinet, E., Géraud, T.: Appendix of the present paper (2013), `http://www.lrde.epita.fr/Olena/maxtree`

[3] Hesselink, W.H.: Salembier's min-tree algorithm turned into breadth first search. Information Processing Letters 88(5), 225–229 (2003)

[4] Jones, R.: Connected filtering and segmentation using component trees. Computer Vision and Image Understanding 75(3), 215–228 (1999)

[5] Levillain, R., Géraud, T., Najman, L.: Why and how to design a generic and efficient image processing framework: The case of the Milena library. In: Proc. of ICIP, pp. 1941–1944 (2010), `http://olena.lrde.epita.fr`

[6] Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. IVC 22(10), 761–767 (2004)

[7] 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)

[8] Menotti, D., Najman, L., de Albuquerque Araújo, A.: 1D component tree in linear time and space and its application to gray-level image multithresholding. In: Proc. of Intl. Symp. on Math. Morphology, pp. 437–448 (2007)

[9] Monasse, P., Guichard, F.: Fast computation of a contrast-invariant image representation. IEEE Trans. on Image Processing 9(5), 860–872 (2000)

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

[11] Nistér, D., Stewénius, H.: Linear time maximally stable extremal regions. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 183–196. Springer, Heidelberg (2008)

[12] 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)

[13] Perret, B., Lefevre, S., Collet, C., Slezak, É.: Connected component trees for multivariate image processing and applications in astronomy. In: Proc. of International Conference on Pattern Recognition, pp. 4089–4092 (2010)

[14] Reinders, J.: Intel threading building blocks: outfitting C++ for multi-core processor parallelism. O'Reilly Media, Incorporated (2007)

[15] Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. IEEE Transactions on Image Processing 7(4), 555–570 (1998)

[16] Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Trans. on Ima. Proc. 4(8), 1153–1160 (1995)

[17] Tarjan, R.E.: Efficiency of a good but not linear set union algorithm. Journal of the ACM 22(2), 215–225 (1975)

[18] Urbach, E.R., Wilkinson, M.H.F.: Shape-only granulometries and grey-scale shape filters. In: Proc. of ISMM, pp. 305–314 (2002)

[19] Vincent, L.: Grayscale area openings and closings, their efficient implementation and applications. In: Proc. of EURASIP Workshop on Mathematical Morphology and its Applications to Signal Processing, pp. 22–27 (1993)

[20] 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. on Image Processing 16(12), 2943–2952 (2007)

[21] 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 Trans. on PAMI 30(10), 1800–1813 (2008)

[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)

[23] Wilkinson, M.H.F.: A fast component-tree algorithm for high dynamic-range images and second generation connectivity. In: ICIP, pp. 1021–1024 (2011)

[24] Xu, Y., Géraud, T., Najman, L.: Morphological filtering in shape spaces: Applications using tree-based image representations. In: Proc. of International Conference on Pattern Recognition, pp. 1–4 (2012)

Constructive Links between Some Morphological Hierarchies on Edge-Weighted Graphs[⋆]

Jean Cousty, Laurent Najman, and Benjamin Perret

Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, A3SI, ESIEE
{j.cousty,l.najman,b.perret}@esiee.fr

Abstract. In edge-weighted graphs, we provide a unified presentation of a family of popular morphological hierarchies such as component trees, quasi flat zones, binary partition trees, and hierarchical watersheds. For any hierarchy of this family, we show if (and how) it can be obtained from any other element of the family. In this sense, the main contribution of this paper is the study of all constructive links between these hierarchies.

Introduction

In recent years, (supervised) image segmentations in edge weighted graphs received a lot of attention. In this framework, several methods [1–5] were designed to segment images into partitions made of connected regions that are optimal in the sense of some well-known problems of combinatorial optimization such as min-cuts, random walks, or minimum spanning trees.

Some of these methods (see [1, 6]) also satisfy a "scale consistency property" that assesses the robustness of the detected contours and regions over scales. Given three image seed points x, y, and z that mark three objects of interest, a segmentation S into three regions obtained from the three seeds x, y and z (*i.e.*, each region contains one seed) "is consistent" with a segmentation S' into two regions obtained from the two seeds x and y if when a pixel belongs to the region of a seed in S, then it necessarily belongs to the region of S' that contains this seed. More generally, a segmentation is called hierarchical if it defines segmentations at different detail levels such that the segmentations at coarser levels can be obtained from those at finer levels by simple merge operations.

In fact, hierarchical segmentation methods are not limited to edge-weighted graphs (see *e.g.*, [7–10]). In particular, in mathematical morphology, component trees [11], quasi-flat zones [12, 13], binary partition trees [14] and watersheds [15–17] are hierarchies at the basis of efficient segmentation and filtering methods.

In this paper, we study these morphological hierarchies defined from edge-weighted graphs, and we provide a unified presentation of this family. For any hierarchy of this family, we show if (and how) it can be obtained from any

[⋆] This work received funding from the Agence Nationale de la Recherche, contract ANR-2010-BLAN-0205-03.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 86–97, 2013.

Table 1. Summary of the main results. In the table, T stands for any minimum spanning tree of G, $\mathcal{S}$ stands for any sequence of minima of F, $\mathcal{PH}(G)$ is the partition-hierarchy of G, $\mathcal{MH}(G)$ is the min-hierarchy of G, $\mathcal{PH}(T)$ is the partition-hierarchy of T, $\mathcal{MH}(T)$ is the min-hierarchy of T, $\mathcal{Q}$ is the quasi-flat zones hierarchy, $\mathcal{B}_{\prec}$ is the binary partition hierarchy by the ordering $\prec$, and $\mathcal{H}_{S}$ is an MSF hierarchy for $\mathcal{S}$. In a cell, the symbol $\Longleftarrow$ (resp. $\Longrightarrow$) indicates that the hierarchy corresponding to the column (resp. line) of the cell can be obtained from the one corresponding to the line (resp. column) of the cell, and the symbol $\Longleftrightarrow$ (resp. $\times$) indicates that two hierarchies can be (resp. cannot be) obtained one from each other.

	$\mathcal{PH}(G)$	$\mathcal{MH}(G)$	$\mathcal{PH}(T)$	$\mathcal{MH}(T)$	$\mathcal{Q}$	$\mathcal{B}_{\prec}$	$\mathcal{H}_{S}$
$\mathcal{PH}(G)$	$\Longleftrightarrow$	$\Longleftrightarrow$	$\Longrightarrow$	$\Longrightarrow$	$\Longrightarrow$	$\times$	$\times$
$\mathcal{MH}(G)$	$\Longleftrightarrow$	$\Longleftrightarrow$	$\Longrightarrow$	$\Longrightarrow$	$\Longrightarrow$	$\times$	$\times$
$\mathcal{PH}(T)$	$\Longleftarrow$	$\Longleftarrow$	$\Longleftrightarrow$	$\Longleftrightarrow$	$\Longrightarrow$	$\Longleftarrow$	$\times$
$\mathcal{MH}(T)$	$\Longleftarrow$	$\Longleftarrow$	$\Longleftrightarrow$	$\Longleftrightarrow$	$\Longrightarrow$	$\Longleftarrow$	$\times$
$\mathcal{Q}$	$\Longleftarrow$	$\Longleftarrow$	$\Longleftarrow$	$\Longleftarrow$	$\Longleftrightarrow$	$\Longrightarrow$	$\times$
$\mathcal{B}_{\prec}$	$\times$	$\times$	$\Longrightarrow$	$\Longrightarrow$	$\Longrightarrow$	$\Longleftrightarrow$	$\Longrightarrow$
$\mathcal{H}_{S}$	$\times$	$\times$	$\times$	$\times$	$\times$	$\Longleftarrow$	$\Longleftrightarrow$

other element of the family. In this sense, the main contribution of this paper is the study of all constructive links between these morphological hierarchies. For establishing these links, the minimum spanning trees play a central role. Table 1 indicates all links that are shown in this paper. An important consequence of our results is the design of efficient algorithms based on Kruskal minimum spanning tree algorithm to compute these morphological hierarchies in quasi linear-time. These algorithms are presented in [18].

1 Graphs

We define a *graph* as a pair $X = (V(X), E(X))$ where $V(X)$ is a finite set and $E(X)$ is composed of unordered pairs of distinct elements in $V(X)$, *i.e.*, $E(X)$ is a subset of $\{\{x, y\} \subseteq V(X) \mid x \neq y\}$. Each element of $V(X)$ is called a *vertex or a point (of G)*, and each element of $E(X)$ is called an *edge* (of X).

Let X and Y be two graphs. If $V(X) \subseteq V(Y)$ and $E(X) \subseteq E(Y)$, then X and Y are ordered and we write $X \sqsubseteq Y$. If $X \sqsubseteq Y$, we say that X is a *subgraph* of Y, or that X is *smaller* than Y and that Y is *greater* than X. The *intersection of* X *and* Y is the graph $X \sqcap Y = \{V(X) \cap V(Y), E(X) \cap E(Y)\}$ and the *union of* X *and* Y is the graph $X \sqcup Y = \{V(X) \cup V(Y), E(X) \cup E(Y)\}$. The set of all subgraphs of a graph G is denoted by 2^G. The set 2^G equipped with the order relation $\sqsubseteq$ is a lattice whose infimum and supremum are the binary operations $\sqcap$ and $\sqcup$ respectively (see [19] for a morphological study of this lattice).

Let X be a graph. A *path (in X)* is a sequence $(x_1, \ldots, x_n)$ of points of $V(X)$ such that $\{x_i, x_{i+1}\} \in E$ for any i in $[1, n-1]$. A path with no repeated vertex is said to be *simple*. The graph X is *connected* if there exists a path between any two vertices of X. A *(connected) component of* X is a subgraph Y of X that is connected and such that, for any connected graph Z, we have $Y = Z$

whenever the relation $Y \sqsubseteq Z \sqsubseteq X$ holds true. We denote by $\mathcal{CC}(X)$ the set of all components of X and, if x is a vertex in $V(X)$, we denote by $\mathcal{CC}_x(X)$ the unique element of $\mathcal{CC}(X)$ whose vertex set contains x.

Important Notations. In the sequel of this paper, the symbol G denotes a connected graph. Furthermore, to shorten the notations, its vertex and edge sets are denoted by V and E respectively instead of $V(G)$ and $E(G)$.

We finish this section with the presentation of an adjunction that is known for playing the role of a building block for morphology on graphs [19]. It will be useful for expressing several properties in the sequel of this article. We denote by ϵ the operator that maps to any subset X of V the subset of E made of the edges of G composed of two points in X, *i.e.*, $\epsilon(X) = \{\{x, y\} \in E \mid x \in X, y \in X\}$. We denote by δ the operator that maps to any subset X of E the subset of V that contains every vertex in V which belongs to an edge in X, *i.e.*, $\delta(X) = \cup\{\{x, y\} \in X\}$. The pair (ϵ, δ) is an adjunction [19]. Let $V' \subseteq V$ and $E' \subseteq E$. Using usual graph terminology, the graphs $(V', \epsilon(V'))$ and $(\delta(E'), E)$ are called the *graph induced by* V' and the *graph induced by* E' respectively.

2 Partitions and Hierarchies

For segmentation purposes, one is often interested in finding partitions of V. We denote by 2^V the set of all subsets of V. Recall that a subset $\mathcal{V}$ of 2^V whose elements are disjoint and nonempty is a *partial partition (of* V*)*. The union of a partial partition is called its *support*. A *partition (of* V*)* is a partial partition whose support is V.

In the following, subgraphs of G will be used to obtain partitions of V. Let X be a subgraph of G. We denote by $\mathcal{V}_{CC}(X)$ the set that contains the vertex set of every component of X, *i.e.*, $\mathcal{V}_{CC}(X) = \{V(Y) \mid Y \in \mathcal{CC}(X)\}$. Remark that the set $\mathcal{V}_{CC}(X)$ is a partial partition of V whose support is $V(X)$. This partial partition is called the *(partial) partition induced by* X.

A set $\mathcal{H} \subseteq 2^V$ (resp. $\mathcal{H} \subseteq 2^G$) is a *hierarchy on* V *(resp.* G*)* if any two elements of $\mathcal{H}$ are either disjoint or nested, *i.e.*, for any $H_1, H_2 \in \mathcal{H}$, we have $H_1 \cap H_2 \in \{\emptyset, H_1, H_2\}$ (resp. $H_1 \sqcap H_2 \in \{(\emptyset, \emptyset), H_1, H_2\}$). A hierarchy $\mathcal{H}$ on V (resp. G) is *complete* if V (resp. G) is in $\mathcal{H}$ and if for any $v \in V$, we have $\{v\} \in \mathcal{H}$, (resp. $\{(\{v\}, \emptyset)\} \in \mathcal{H}$). It is well-known that the Hasse diagram of a hierarchy (resp. complete hierarchy) is a directed forest (resp. tree), often called the dendrogram of the hierarchy. In practice, this dendrogram is used as a representation of the hierarchy. Let X and Y be two distinct elements of a hierarchy $\mathcal{H}$ (on V or G), following the terminology of the dendrogram, we say that Y *is a child of* X if Y is the largest proper subset of X among the elements of $\mathcal{H}$, *i.e.*, if $Y \subseteq X$, and, for any $Z \in \mathcal{H}$ such that $Y \subseteq Z \subseteq X$, we have $Z = X$ or $Z = Y$. If Y is a child of X, we say that X is the *parent of* Y.

Let $\mathcal{H}$ be a hierarchy on V (resp. G) and let X be an element of $\mathcal{H}$. A *minimum of* $\mathcal{H}$ is an element of $\mathcal{H}$ that has no child. Let $\mathcal{C} \subseteq \mathcal{H}$. We say that $\mathcal{C}$ is a *cut of* $\mathcal{H}$ if i) the elements of $\mathcal{C}$ are pairwise disjoint, and ii) for any minimum M of $\mathcal{H}$, the set $\mathcal{C}$ contains an element that is greater than M. If $\mathcal{H}$ is a hierarchy on V, we say that $\mathcal{H}$ *is a hierarchy of partitions (on V)* whenever any cut of $\mathcal{H}$ is a partition of V. The following property characterizes the hierarchies of partitions from their minima.

Property 1. *Let $\mathcal{H}$ be a hierarchy on V. The hierarchy $\mathcal{H}$ is a hierarchy of partitions if and only if the set of its minima is a partition.*

A direct corollary is that any complete hierarchy on V is a hierarchy of partitions.

The hierarchies on G may be used to obtain hierarchies (of partitions) on V. Let $\mathcal{H}$ be a hierarchy on G. We denote by $\mathcal{V}(\mathcal{H})$ the hierarchy on V defined by $\mathcal{V}(\mathcal{H}) = \{V(X) \mid X \in \mathcal{H}\}$ and we say that $\mathcal{V}(\mathcal{H})$ is the *hierarchy (on V) induced by $\mathcal{H}$*. Observe that the hierarchy $\mathcal{H}$ on G induces a hierarchy of partitions on V if and only if any vertex of G is a vertex of a minimum of $\mathcal{H}$.

Let $\mathcal{H}$ be a hierarchy on V (resp. on G), and let x be in V. The greatest element of $\mathcal{H}$ that contains x (resp. whose vertex set contains x) is denoted by $\mathcal{CC}_x(\mathcal{H})$. Observe that if $\mathcal{H}$ is complete, then $\mathcal{CC}_x(\mathcal{H})$ is exactly V (resp. G).

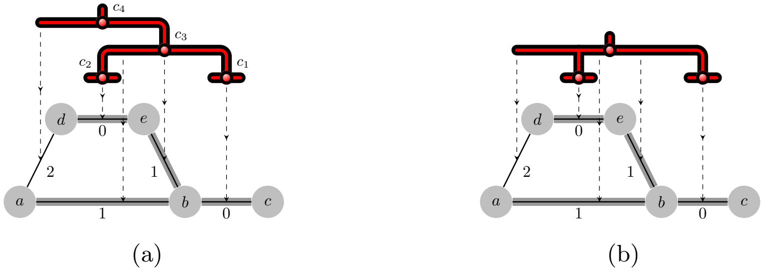
3 Component Trees

Intuitively, component trees [11] may be seen as hierarchies obtained from the connected components of an image. In particular, the min-tree is a well known hierarchical representation that is useful for anti-extensive connected operators. In this expression, the term min is used in reference to the leaves of these trees that are the regional minima of the images. In this section, we provide definitions of regional minima and of min-trees for edge-weighted graphs. Furthermore, on the same basis, we provide a definition of a hierarchy of partitions that allows links to be drawn between min-trees and quasi-flat zones.

Important Notation. In the sequel, we denote by F a function from E to $\mathbb{R}^+$ that weights the edges of E. Therefore, the pair (G, F) is called an *edge-weighted graph*, and, for any $u \in E$, the value $F(u)$ is called the *weight of u*.

Let $k \in \mathbb{R}$. A subgraph X of G is a *minimum of F (at weight k)* if i) X is connected; and ii) k is the weight of any edge of X; and iii) the weight of any edge adjacent to X (*i.e.*, any edge that contains exactly one vertex of X) is strictly greater than k.

In order to define the components of a weight map, the simple thresholding operation is used to produce level sets from which connected components can be considered. For given $\lambda \in \mathbb{R}$ and $X \subseteq E$, the *λ-level set of X* (for F) is the set $\chi_\lambda(X)$ of all edges in X whose value is not greater than λ, *i.e.*, $\chi_\lambda(X) = \{e \in X | F(e) \leq \lambda\}$. From the level set $\chi_\lambda(E)$ of E, two interesting graphs can be derived: the first one, called the *λ-level graph of G*, and denoted by $\chi^E_\lambda(G)$, is defined by $\chi^E_\lambda(G) = (\delta(\chi_\lambda(E)), \chi_\lambda(E))$, and the second one, called the *λ-level spanning graph of G* and denoted by $\chi^V_\lambda(G)$, is defined by $\chi^V_\lambda(G) = (V, \chi_\lambda(E))$. More generally, if $X \sqsubseteq G$, the *λ-level graph of X* and the *λ-level spanning graph of X* are defined by $\chi^E_\lambda(X) = (\delta(\chi_\lambda(E(X))), \chi_\lambda(E(X)))$, and $\chi^V_\lambda(X) = (V, \chi_\lambda(E(X)))$ respectively.



(a) (b)

Fig. 1. Illustration of the min-hierarchy of a graph (a) and of its unique minimum spanning tree (b), which is represented by wide edges

Note that $\chi^E_\lambda(G)$ can be derived from $\chi^V_\lambda(G)$ by removing all isolated points of $\chi^V_\lambda(G)$, and that, conversely, $\chi^V_\lambda(G)$ can be derived from $\chi^E_\lambda(G)$ by adding all elements of V to the vertex set of $\chi^E_\lambda(G)$. Hence, we always have $\chi^E_\lambda(G) \sqsubseteq \chi^V_\lambda(G)$. Note also that the partial partition induced by $\chi^V_\lambda(G)$ is always a partition of V whereas the one induced by $\chi^E_\lambda(G)$ is in general not a partition (*i.e.*, its support is in general a proper subset of V).

Definition 2. *Let $X \sqsubseteq G$. The* partition-hierarchy of X (for F)*, denoted by $\mathcal{PH}(X)$, is the set $\mathcal{PH}(X) = \cup\{\mathcal{CC}(\chi^V_\lambda(X)) \mid \lambda \in \mathbb{R}\}$ and the* min-hierarchy of X (for F)*, denoted by $\mathcal{MH}(X)$, is the set $\mathcal{MH}(X) = \cup\{\mathcal{CC}(\chi^E_\lambda(X)) \mid \lambda \in \mathbb{R}\}$.*

The Hasse diagram of the min-hierarchy of G is known as the *min-tree of* (G, F). Fig. 1a shows in red the min-tree of the edge-weighted graph represented in gray. The elements C_1, C_2, C_3 and C_4 of this min-hierarchy (*i.e.*, the nodes of the min-tree) are represented by red horizontal lines. Observe that C_1 and C_2, which are the two components of the 0-level graphs of G, are the graphs induced by $\{\{b, c\}\}$ and $\{\{d, e\}\}$ respectively. The component C_3 (resp. C_4) is the unique connected component of the 1-level graph (resp. 2-level graph) of G; C_3 is the graph induced by $\{\{a, b\}, \{b, c\}, \{b, e\}, \{d, e\}\}$ and C_4 is the graph G itself. The partition-hierarchy of G is a superset of this min-hierarchy, which furthermore contains any subgraph of G made of a single vertex. More generally, as assessed by the following property, the min-hierarchy and the partition-hierarchy of G can always be obtained one from each other. Therefore, the min-hierarchy and the partition-hierarchy of G are equivalent as well as the min-tree of (G, F) and the Hasse diagram of the partition-hierarchy of G.

Property 3. *The min-hierarchy of any subgraph X of G can be obtained by removing from the partition-hierarchy of X the graphs made of a single vertex,* i.e.*, $\mathcal{MH}(X) = \mathcal{PH}(X) \setminus \{(\{x\}, \emptyset) \mid x \in V\}$. Conversely, the partition-hierarchy of any subgraph X of G can be obtained by adding to the min-hierarchy of X all graphs made of a single vertex,* i.e.*, $\mathcal{PH}(X) = \mathcal{MH}(X) \cup \{(\{x\}, \emptyset) \mid x \in V\}$.*

Observe that the partition-hierarchy of G indeed induces a hierarchy of partitions on V, whereas, in general, the min-hierarchy of G does not. In the next section, we will study the minimum spanning trees of G, and we will see that these particular subgraphs of G are sufficient to recover the hierarchies of partitions induced by the min-hierarchy and the partition-hierarchy of G.

4 Minimum Spanning Trees

The minimum spanning tree is a typical and well-known problem of combinatorial optimization. It has been applied for many years to image analysis problems. The main result of this section states that the hierarchy of partitions induced by the partition-hierarchy of any minimum spanning tree of G is exactly the same as the hierarchy of partitions induced by the partition-hierarchy of the graph G itself. Furthermore, the minimum spanning trees are minimal (with respect to the relation $\sqsubseteq$) for this property.

A graph X *is spanning (for G)* if $V(X) = V$. Let $X \sqsubseteq G$. *The weight of* X *(for F)*, denoted by $F(X)$, is the sum of the weights of the edges in $E(X)$: $F(X) = \sum_{u \in E(X)} F(u)$. A connected spanning graph T is a *minimum spanning tree (of* (G, F) if the weight of T is less than or equal to the weight of any other connected graph that is spanning.

Property 4. *Let T be any minimum spanning tree of G. Then, the partitions induced by $\chi^V_\lambda(T)$ and by $\chi^V_\lambda(G)$ are the same.*

Let $X \sqsubseteq G$. We denote by $\phi(X)$ the graph induced by the vertex set of X:$\phi(X) = (V(X), \epsilon(V(X)))$. Note that ϕ is both a dilation and a closing in the lattice 2^G of all subgraphs of G (for more details, see [19] where ϕ is denoted by α_2). For a given hierarchy $\mathcal{H}$ of graphs, we write $\varphi(\mathcal{H}) = \{\phi(X) \mid X \in \mathcal{H}\}$. It can be seen that $V(\phi(X)) = V(X)$. Thus the hierarchies on V induced by $\mathcal{H}$ and $\varphi(\mathcal{H})$ are the same, *i.e.*, we always have $\mathcal{V}(\varphi(\mathcal{H})) = \mathcal{V}(\mathcal{H})$.

Given two hierarchies $\mathcal{H}_1$ and $\mathcal{H}_2$ whose elements are ordered by the relations $\leq_1$ and $\leq_2$ respectively, an *(order) isomorphism from* $\mathcal{H}_1$ *to* $\mathcal{H}_2$ is a bijection f from $\mathcal{H}_1$ to $\mathcal{H}_2$ such that for any $X, Y \in \mathcal{H}_1$, $X \leq_1 Y$ if and only if $f(X) \leq_2 f(Y)$. If there exists an isomorphism from $\mathcal{H}_1$ to $\mathcal{H}_2$, then $\mathcal{H}_1$ and $\mathcal{H}_2$ are said *isomorphic* and we write $\mathcal{H}_1 \cong \mathcal{H}_2$. Note that two hierarchies that are isomorphic can be represented by the same Hasse diagram.

Property 5. *Let T be any minimum spanning tree of G. Then the two following statements hold true:*

1. $\mathcal{PH}(T) \cong \varphi(\mathcal{PH}(G))$*; and*
2. $\mathcal{V}(\mathcal{PH}(T)) = \mathcal{V}(\varphi(\mathcal{PH}(G))) = \mathcal{V}(\mathcal{PH}(G))$.

In other words, the hierarchies induced by the partition-hierarchy of a minimum spanning tree and by the graph itself are the same. Furthermore, due to the mapping φ, the partition-hierarchy of any minimum spanning tree of G can be recovered from the partition hierarchy of G. On the contrary, the converse is in general not true. Hence, in general, there is more information in the partition-hierarchy of G than in partition-hierarchy of any of its minimum spanning trees. When available, such information may be used for further processing.

Property 5 is illustrated on the edge-weighted graph (G, F) of Fig. 1, where the red trees in (a) and (b) represent respectively $\mathcal{PH}(G)$ and $\mathcal{PH}(T) \cong \varphi(\mathcal{PH}(G))$, T being the minimum spanning tree depicted with "wide" edges.

5 Quasi-flat Zones

The quasi-flat zones (see *e.g.* [8, 12, 13]) have been studied since the 70's, and they have been used recently as a basis for constrained connectivity segmentations. In this section, we investigate the links between the quasi-flat zones, the min-trees, the partition-hierarchies of a graph G and of its minimum spanning trees.

Let $\lambda \in \mathbb{R}$. A path $\pi = (x_0, \ldots, x_n)$ is λ-connected if for any i in $[\![0, n-1]\!] = \{0, \ldots n-1\}$, we have $\{x_i, x_{i+1}\} \in E$ and $F(\{x_i, x_{i+1}\}) \leq \lambda$. For any two vertices x and y in V, we set $\lambda - \Pi(x, y)$ as the set of all λ-connected paths from x to y. The *λ-flat zone (or quasi-flat zone at level λ) of a vertex x* is the set $\lambda - CC(x) = \cup\{y \in V \mid \lambda - \Pi(x, y) \neq \emptyset\}$. The set $Q_\lambda = \{\lambda - CC(x) \mid x \in V\}$ of λ-flat zones over all vertices in E is a partition.

Definition 6. *The set $\mathcal{Q} = \cup\{Q_\lambda \mid \lambda \in \mathbb{R}\}$ is the* quasi-flat zones hierarchy *of F.*

The quasi-flat zones hierarchy is a complete hierarchy, and thus also a hierarchy of partitions. In the literature, the term α-tree was coined by G. Ouzounis and P. Soille for the Hasse diagram of the quasi-flat zones hierarchy [20].

For any $\lambda \in \mathbb{R}$, it can be seen that Q_λ is the partition induced by the λ-level graph $\chi^V_\lambda(G)$ of G. Hence, by Property 4, the partition Q_λ is also the partition induced by the λ-level graph of any minimum spanning tree of G. Therefore, the following property linking the quasi-flat zones hierarchy to the partition-hierarchies of any minimum spanning tree of G can be established.

Property 7. *Let T be a minimum spanning tree of G. Then, the two following statements hold true:*

1. *$Q \cong \mathcal{PH}(T)$; and*
2. *$Q = \mathcal{V}(\mathcal{PH}(T))$.*

The first relation states that the quasi-flat zones hierarchy and the partition-hierarchy of T are isomorphic. Due to Property 5.1, we deduce that these two hierarchies are also isomorphic to $\varphi(\mathcal{PH}(G))$ obtained by simplifying the partition-hierarchy of G. Furthermore, by Property 3, we deduce that these two hierarchies may also be obtained form the min-hierarchy of G. Hence, Property 7.1 states that the α-tree and the partition-tree of any minimum spanning tree of G are the same and that they both can be obtained from the partition- and min-trees of the graph G itself. The second relation states that the quasi-flat zones hierarchy is exactly the hierarchy of partitions induced by the partition-hierarchy of any minimum spanning tree of G. It thus states how to obtain the quasi flat zones hierarchy from any other hierarchy previously presented in this paper.

6 Binary Partition Trees

In this section, we present the binary partition hierarchies by (altitude) orderings. These hierarchies fall into the wide category of binary partition trees as introduced by P. Salembier [21]. Then, we state that the quasi-flat zones hierarchy can be recovered from this hierarchy, and we show a mapping from the elements of these hierarchies to the edge-set of the minimum spanning trees of (G, F). Note that Meyer studied similar links between catchment basins and minimum spanning trees in [17].

Let $\prec$ be a total ordering on E, *i.e.*, $\prec$ is a binary relation that is transitive and trichotomous (for any u and v in E only one of the relations $u \prec v$, $v \prec u$, and $u = v$ holds true). Let k be any element in $[\![1, |E|]\!]$, we denote by $u_k^{\prec}$ the k-th element of E with respect to $\prec$.

Definition 8. *Let k be an element in $[\![1, |E|]\!]$. We set $\mathcal{B}^0 = \big\{\{x\} \mid x \in V\big\}$. The* partial binary partition hierarchy $\mathcal{B}^k$ at rank k (by the ordering $\prec$) *is the hierarchy on V defined by $\mathcal{B}^k = \mathcal{B}^{k-1} \cup \big\{\mathcal{CC}_x(\mathcal{B}^{k-1}) \cup \mathcal{CC}_y(\mathcal{B}^{k-1})\big\}$ where $u_k^{\prec} = \{x, y\}$.*

The partial binary partition hierarchy at rank $|E|$ is called the binary partition hierarchy by (the ordering) $\prec$ *and it is denoted by $\mathcal{B}_{\prec}$.*

The Hasse diagram of the binary partition hierarchy is known in the literature as the binary partition tree (see, *e.g.*, Fig. 2a). Note that, for every possible value of k, the partial binary partition hierarchy at rank k is a hierarchy of partitions and furthermore the binary partition hierarchy is a complete hierarchy.

Let $\prec$ be an ordering on E, and let $k \in [\![1, |E|]\!]$. Observe that the partial binary partition hierarchy at rank k is equal to the partial binary partition hierarchy at rank $k - 1$ if and only if the k-th edge for $\prec$ links two vertices that are already in a same set of $\mathcal{B}^{k-1}$ (see *e.g.* the hierarchies $\mathcal{B}^6$ and $\mathcal{B}^7$ in Fig. 2a). Hence, we may associate to any element X in $\mathcal{B}_{\prec}^{\star} = \mathcal{B}_{\prec} \setminus \mathcal{B}^0$ the lowest rank at which a partial binary partition tree contains X. This rank is called the *rank of* X, it is denoted by $r(X)$ and we have $r(X) = \min\{k \in [\![1, n]\!] \mid X \in \mathcal{B}^k\}$. This rank also allows us to directly map the elements of $\mathcal{B}_{\prec}^{\star}$ to a subset of E. Let $X \in \mathcal{B}_{\prec}^{\star}$, the *building edge of* X is the $r(X)$-th edge of E for $\prec$. The set of building edges of all elements in $\mathcal{B}_{\prec}^{\star}$ is called the *building set of* $\mathcal{B}_{\prec}$.

We say that an ordering $\prec$ on E is an *altitude ordering (for F)* if $F(u) \leq F(v)$ for any two u and v in E such that $u \prec v$. If there is only one altitude ordering for F, then we say that F *is totally ordering.*

Property 9. *Let $\prec$ be an altitude ordering, and let $\mathcal{B}_{\prec}$ be the binary partition hierarchy by $\prec$. If F is totally ordering, then the two following statements hold true:*

1. *the graph induced by the building set of $\mathcal{B}_{\prec}$ is the unique minimum spanning tree of F; and*
2. *$\mathcal{B}_{\prec} = \mathcal{Q}$.*

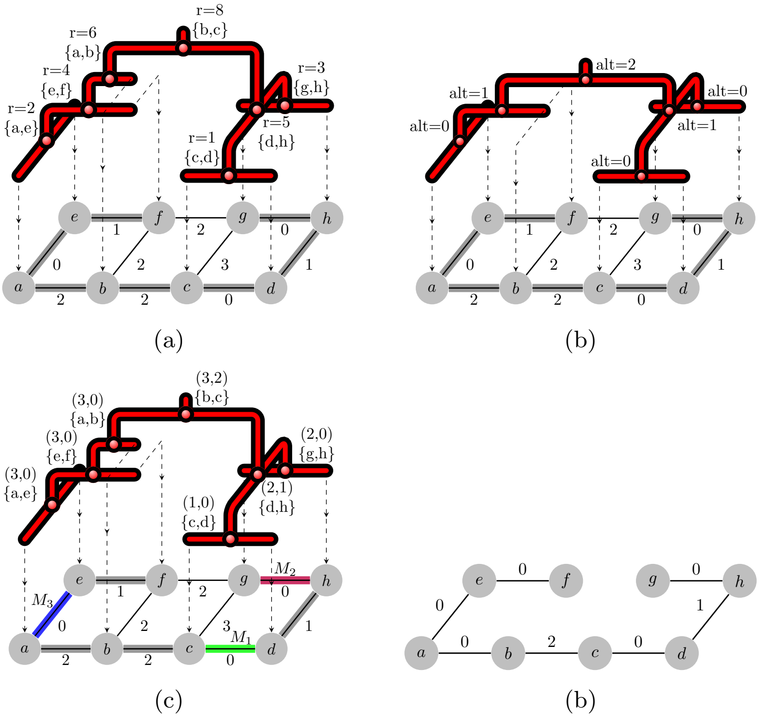


Fig. 2. a: A binary partition hierarchy/tree $\mathcal{B}_{\prec}$ for the altitude ordering $\{c, d\} \prec \{a, e\} \prec \{g, h\} \prec \{e, f\} \prec \{d, h\} \prec \{a, b\} \prec \{b, f\} \prec \{b, c\} \prec \{f, g\} \prec \{c, g\}$. b: $\Psi(\mathcal{B}_{\prec})$. c: The same binary partition hierarchy with a sequence $\mathcal{S} = \langle M_1, M_2, M_3 \rangle$ of minima of F and, for each non-leaf element, a pair of values $(ext, pers)$ made of the extinction of the component and the persistence of its building edge (note that the extinction of a leaf/singleton is always 0). d: The edge-weighted graph whose partition-hierarchy is the hierarchy induced by $\prec$ and $\mathcal{S}$, which is thus also the MSF hierarchy for $\mathcal{S}$.

Hence, under the conditions of the previous property, we deduce from Properties 7 and 5 that $\mathcal{B}_{\prec}$ is also isomorphic to $\mathcal{PH}(T)$ and to $\varphi(\mathcal{PH}(G))$. Thus, it can be obtained as a simplification of the partition-hierarchy $\mathcal{PH}(G)$ and also, by Property 3, as a simplification of the min-hierarchy $\mathcal{MH}(G)$.

Let $\prec$ be an altitude ordering. Let $X \in \mathcal{B}^{\star}_{\prec}$, we call *altitude of* X, the weight of its building edge. We say that X *is principal for* $\mathcal{B}_{\prec}$ if it has no parent or if its altitude is less than the one of its parent. The set of all principal elements of $\mathcal{B}_{\prec}$ is denoted $\Psi(\mathcal{B}_{\prec})$ (see Figs. 2a and b for illustrations).

Property 10. *Let $\prec$ be an altitude ordering, and let $\mathcal{B}_{\prec}$ be the binary partition hierarchy by $\prec$. Then, the two following statements hold true:*

1. *the graph induced by the building set of $\mathcal{B}_{\prec}$ is a minimum spanning tree of F; and*
2. *$\mathcal{Q} = \Psi(\mathcal{B}_{\prec})$.*

The previous property states that the quasi-flat zones hierarchy $\mathcal{Q}$ can be obtained by simplifying the binary partition hierarchy $\mathcal{B}_{\prec}$. In fact, contrarily to the case of maps which are totally ordering, the converse is, in general, not true: the binary partition hierarchy cannot be obtained from the quasi-flat zones hierarchy or from the partition-hierarchy of a minimum spanning tree. Furthermore, it can be shown that, in general, one cannot recover a binary partition hierarchy from a min-/partition-hierarchy of G either.

7 Hierarchies of Minimum Spanning Forests

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, is known to be equivalent to the one of minimum spanning tree. Then, hierarchies of minimum spanning forests are introduced. Each such hierarchy induces a hierarchy of partitions on V. Finally, we state the main result of this section that shows how hierarchies of minimum spanning forests can be obtained from binary partition hierarchies.

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 i) Y is spanning; ii) Y is rooted in X; and iii) 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 and rooted in X). Furthermore, any minimum spanning tree of G is called an MSF rooted in the empty graph.

For instance, the graphs induced by the null edges of Fig. 2d is an MSF rooted in the graph made of the minima M_1, M_2 and M_3 shown in Fig. 2c.

A possible definition for watershed, called *watershed-cuts*, follows the drop of water principle. In [4], 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. The next definition presents a notion of hierarchy of MSFs rooted in such subsets.

We denote by $\mathcal{M}_F$ the set of all minima of F.

Definition 11 (MSF hierarchy, [6]). *Let* $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ *be a sequence of pairwise distinct minima of* F *and let* $\langle X_0, \ldots, X_\ell \rangle$ *be a sequence of subgraphs of* G *such that:*

1. *for any* $i \in [\![0, \ell]\!]$, 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* $X_{i-1} \sqsubseteq X_i$*.*

The set $\mathcal{T} = \cup\{CC(X_i) \mid i \in [\![0, \ell]\!]\}$ *is called an* MSF hierarchy for $\mathcal{S}$*.*

Let $\prec$ be an altitude ordering on E, let $\mathcal{S} = \langle M_1, \ldots, M_\ell \rangle$ be a sequence of pairwise distinct minima of F, and let $X \in \mathcal{B}_{\prec}$. The *extinction value of* X *for* $\mathcal{S}$ is 0 if there is no element of $\mathcal{S}$ whose vertex set is included in X, or, otherwise, it is set to the highest index k such that the vertex set of M_k is included in X.

Intuitively, if we see the sequence $\mathcal{S}$ as a sequence of "markers" ranked by increasing "importance", the extinction value of a set X in $\mathcal{H}$ can be seen as the rank of the most important marker of X (*i.e.*, that is contained in X). For instance, in Fig. 2, the extinction value of every component of the binary partition tree is given for the sequence $\langle M_1, M_2, M_3\rangle$, where M_1 (resp. M_2 and M_3) is the minimum induced by $\{\{c, d\}\}$ (resp. $\{\{g, h\}\}$ and $\{\{e, f\}\}$). Dually, one can intuitively consider the persistence of an edge u as the highest rank k such that the vertices linked by u belong to distinct regions of the partitions obtained by considering only the k most important markers (see *e.g.* Figs. 2c, and d). Based on this notion of persistence, Property 12 states that MSF hierarchies can be obtained using only the binary partition trees by altitude orderings.

Let $\prec$ be an altitude ordering on E. Let $\mathcal{S} = \langle M_1, \ldots, M_\ell\rangle$ be a sequence of pairwise distinct minima of F. Let u be an edge in the building set of $\mathcal{B}_\prec$, and let X be the unique element in $\mathcal{B}_\prec$ whose building edge is u. The *persistence value of* u is the minimum of the extinction values of the children of X. Let $i \in [\![1, \ell]\!]$. We denote by B_i the set of building edges whose persistence value is lower than or equal to i and the set of graphs $\cup\{\mathcal{CC}((V, B_i)) \mid i \in [\![1, \ell - 1]\!]\}$ is called the *hierarchy induced by* $\prec$ *and* $\mathcal{S}$.

Property 12. *Let $\mathcal{S} = \langle M_1, \ldots, M_\ell\rangle$ be a sequence of pairwise distinct minima of F and let $\mathcal{T}$ be a hierarchy on G. The hierarchy $\mathcal{T}$ is an MSF hierarchy for $\mathcal{S}$ if and only if there exists an altitude ordering $\prec$ such that $\mathcal{T}$ is the hierarchy induced by $\prec$ and $\mathcal{S}$.*

Conclusion

This paper investigates the links between some popular morphological hierarchies. Table 1 sums up the links shown in this paper. These links open the way towards a family of efficient algorithms, based on Kruskal minimum spanning tree algorithms, for computing morphological hierarchies. These algorithms are presented in the companion paper [18]. Furthermore, the links established in this paper invites us to bridge hierarchical processing coming from different family of hierarchies. Evaluating the impact of mixing these techniques is left for future work. It also allows for designing new hierarchical methods derived from image predicate which are not necessarily hierarchical (see a first example in [22]). Finally, the links between the hierarchical methods presented in this paper and those based on self-dual tree of level lines [23] still need to be investigated.

References

1. Morris, O.J., de Lee, M.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)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23(11), 1222–1239 (2001)

3. Grady, L.: Random walks for image segmentation. PAMI 28(11), 1768–1783 (2006)
4. 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)
5. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power Watersheds: A Unifying Graph Based Optimization Framework. PAMI 33(7), 1384–1399 (2011)
6. Cousty, J., Najman, L.: Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 272–283. Springer, Heidelberg (2011)
7. Guigues, L., Cocquerez, J.P., Men, H.L.: Scale-sets image analysis. IJCV 68(3), 289–317 (2006)
8. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. PAMI 30(7), 1132–1145 (2008)
9. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. PAMI 33(5), 898–916 (2011)
10. Pont-Tuset, J., Marques, F.: Supervised assessment of segmentation hierarchies. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 814–827. Springer, Heidelberg (2012)
11. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. TIP 7(4), 555–570 (1998)
12. Nagao, M., Matsuyama, T., Ikeda, Y.: Region extraction and shape analysis in aerial photographs. CGIP 10(3), 195–223 (1979)
13. 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)
14. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. TIP 9(4), 561–576 (2000)
15. Beucher, S.: Watershed, hierarchical segmentation and waterfall algorithm. In: ISMM, pp. 69–76 (1994)
16. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. PAMI 18(12), 1163–1173 (1996)
17. Meyer, F.: The dynamics of minima and contours. In: ISMM, pp. 329–336 (1996)
18. Najman, L., Cousty, J., Perret, B.: Playing with kruskal: algorithms for morphological trees in edge-weighted graphs. In: Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 135–146. Springer, Heidelberg (2013)
19. Cousty, J., Najman, L., Dias, F., Serra, J.: Morphological filtering on graphs. CVIU 117, 370–385 (2012)
20. 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)
21. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. TIP 9(4), 561–576 (2000)
22. Guimarães, S.J.F., Cousty, J., Kenmochi, Y., Najman, L.: A hierarchical image segmentation algorithm based on an observation scale. In: Gimel'farb, G., Hancock, E., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., Yamada, K. (eds.) SSPR & SPR 2012. LNCS, vol. 7626, pp. 116–125. Springer, Heidelberg (2012)
23. Caselles, V., Monasse, P.: Geometric Description of Images as Topographic Maps. LNCS, vol. 1984. Springer (2010)

A Quasi-linear Algorithm to Compute the Tree of Shapes of nD Images

Thierry Géraud[1,2], Edwin Carlinet[1,2], Sébastien Crozet[1], and Laurent Najman[2]

[1] EPITA Research and Development Laboratory (LRDE)
[2] Université Paris-Est, LIGM, Équipe A3SI, ESIEE
firstname.lastname@lrde.epita.fr, l.najman@esiee.fr

Abstract. To compute the morphological self-dual representation of images, namely the tree of shapes, the state-of-the-art algorithms do not have a satisfactory time complexity. Furthermore the proposed algorithms are only effective for 2D images and they are far from being simple to implement. That is really penalizing since a self-dual representation of images is a structure that gives rise to many powerful operators and applications, and that could be very useful for 3D images. In this paper we propose a simple-to-write algorithm to compute the tree of shapes; it works for nD images and has a quasi-linear complexity when data quantization is low, typically 12 bits or less. To get that result, this paper introduces a novel representation of images that has some amazing properties of continuity, while remaining discrete.

1 Introduction

The tree of shapes [16] is an important morphological structure that represents images in a self-dual way. Shortly put it can be seen as the result of merging the pair of dual component trees, min-tree and max-tree, into a single tree. Using the tree of shapes has many advantages. Since it is self-dual, it makes no assumption about the contrast of objects (either light object over dark background or the contrary). We only have one structure that represents the image contents so we do not have to juggle with the couple of dual trees. It intrinsically eliminates the redundancy of information contained in those trees. Last, it encodes the spatial inclusion of connected components in gray-level images so it is complementary to some other representations that focus on component (or region) adjacency. As a consequence the tree of shapes is not only an easy access to self-dual operators such as grain filters but it has many applications, as listed in [14] (pp. 15–17), and some very recent works illustrate several powerful perspectives offered by that tree (see [20,21,22], and their bibliography).

In the following we consider a nD digital image u as a function defined on a regular cubical grid (precisely, $u : \mathbb{Z}^n \to \mathbb{Z}$), and to properly deal with some subsets of $\mathbb{Z}^n$ and with their complementary, we consider the dual connectivities c_{2n} and c_{3^n-1}. For any $\lambda \in \mathbb{Z}$, the lower (strict) cuts[1] and upper (large) cuts

[1] We can indifferently use the term "cut" or "threshold".

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 98–110, 2013.

of u are defined as $[\,u < \lambda\,] = \{\,x \in X \mid u(x) < \lambda\,\}$ and $[\,u \geq \lambda\,] = \{\,x \in X \mid u(x) \geq \lambda\,\}$. From them we deduce two sets, $\mathcal{T}_<(u)$ and $\mathcal{T}_\geq(u)$, composed of the connected components of respectively lower and upper cuts of u: $\mathcal{T}_<(u) = \{\,\Gamma \in \mathcal{CC}_{c_{2n}}([\,u < \lambda\,])\,\}_\lambda$ and $\mathcal{T}_\geq(u) = \{\,\Gamma \in \mathcal{CC}_{c_{3^n-1}}([\,u \geq \lambda\,])\,\}_\lambda$, where $\mathcal{CC}$ denotes the operator that gives the set of connected components of a set. The elements of $\mathcal{T}_<(u)$ and $\mathcal{T}_\geq(u)$ respectively give rise to two dual trees: the min-tree and the max-tree of u. We then define two other sets, $\mathcal{S}_<(u)$ (set of lower shapes) and $\mathcal{S}_\geq(u)$ (set of upper shape), as the sets of components of resp. $\mathcal{T}_<(u)$ and $\mathcal{T}_\geq(u)$ after having filled the cavities [2] of those components. With the cavity-filling (or saturation) operator denoted by Sat, we have: $\mathcal{S}_<(u) = \{\,\mathrm{Sat}_{c_{3^n-1}}(\Gamma);\ \Gamma \in \mathcal{T}_<(u)\,\}$ and $\mathcal{S}_\geq(u) = \{\,\mathrm{Sat}_{c_{2n}}(\Gamma);\ \Gamma \in \mathcal{T}_\geq(u)\,\}$.

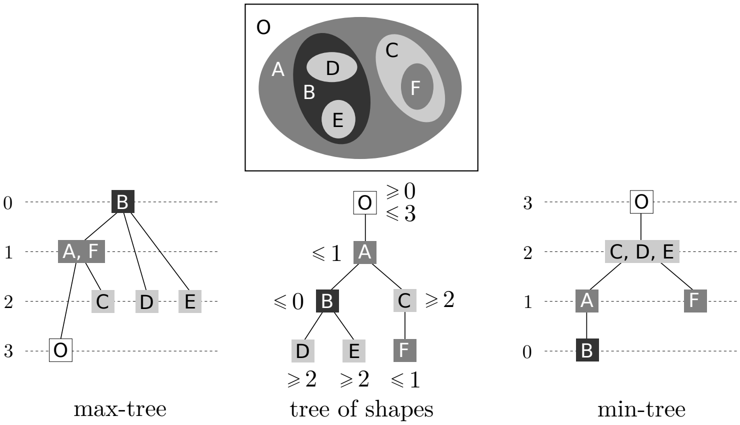


Fig. 1. Three morphological trees of the same image

The set of all shapes $\mathfrak{S}(u) = \mathcal{S}_<(u) \cup \mathcal{S}_\geq(u)$ forms a tree, the so-called *tree of shapes* of u [16]. Indeed, for any pair of shapes Γ and Γ' in $\mathfrak{S}$, we have $\Gamma \subset \Gamma'$ or $\Gamma' \subset \Gamma$ or $\Gamma \cap \Gamma' = \emptyset$. Actually, the shapes are the cavities of the elements of $\mathcal{T}_<$ and $\mathcal{T}_\geq$. For instance, if we consider a lower component $\Gamma \in [\,u < \lambda\,]$ and a cavity H of Γ, this cavity is an upper shape, i.e., $H \in \mathcal{S}_\geq$. Furthermore, in a discrete setting, H is obtained after having filled the cavities of a component of $[\,u \geq \lambda\,]$. Figure 1 depicts on a sample image the three components trees ($\mathcal{T}_<$, $\mathcal{T}_\geq$, and $\mathfrak{S}$). Just note that the Equations so far rely on the pair of dual connectivities, c_{2n} and c_{3^n-1}, so discrete topological problems are avoided, and, in addition, we are forced to consider two kind of cuts: strict ones for c_{2n} and large ones for c_{3^n-1}.

The state-of-the-art of tree of shapes computation (detailed in Section 5) suffers from two major flaws: existing algorithms have a time complexity of

[2] In 2D, a cavity of a set $S \in \Omega$ is called a "hole"; in nD, it is a connected component of $\Omega \backslash S$ which is not the "exterior" of S. Browsing the elements of S in nD, with $n \geq 3$, does not allow to know whether S has a cavity or not[8].

$O(n^2)$ and they cannot easily be extended to nD images. Briefly put, this is due to the fact that either they follow shape contours or they have to know if a component has a cavity[2]. This paper presents an algorithm that can compute the tree of shapes with quasi-linear time complexity when image data are low quantized; furthermore this algorithm straightforwardly applies to nD images.

This paper is organized as follows. First we explain that a well-known algorithmic scheme can be reused to compute the tree of shapes (Section 2). Then this paper introduces a new discrete representation of images (Section 3) that has some properties borrowed from the continuous world. At that point we are ready to glue together the algorithmic scheme and the novel image representation to present a quasi-linear algorithm that compute the tree of shapes (Section 4). Related works about that tree computation is presented so that the reader can compare our approach to existing ones (Section 5). Last, we give a short conclusion (Section 6) [3].

2 Algorithmic Scheme and the Need for Continuity

This section shows that the max-tree algorithm presented in [2] is actually an algorithmic “canvas” [7], that is, a kind of meta-algorithm that can be “filled in” so that it can serve different aims. In the present paper it gives an algorithm to compute the tree of shapes.

2.1 About Union-Find and Component Trees

An extremely simple union-find structure (attributed by Aho to McIlroy and Morris) was shown by Tarjan [19] to be very efficient. This structure, also called disjoint-set data structure or merge-find set, has many advantages that are detailed in [3]; amongst them, memory compactedness, simplicity of use, and versatility. This structure and its related algorithms are of prime importance to handle connected operators [13,6].

Let us denote by $\mathcal{R}$ the ancestor relationship in trees: we have $a\,\mathcal{R}\,p$ iff a is an ancestor of p. $\mathcal{R}$ can be encoded as an array of elements (nodes) so that $a\,\mathcal{R}\,p \Leftrightarrow \mathit{index}_{\mathcal{R}}(a) < \mathit{index}_{\mathcal{R}}(p)$; browsing that array thus corresponds to a downwards browsing of the tree, i.e., from root to leaves. To construct the max-tree of a given image, we rely on a rooted tree defined by a parenthood function,

[3] Due to limited place, this paper does not contain the following topics (they will be included into an extended version of this paper). *A comparison of execution times of existing algorithms.* Actually it is possible to reduce the space complexity (i.e., memory usage) of the algorithm proposed in this paper so the shorter version presented here is not our “competitive” version. *The union-by-rank procedure that guaranties quasi-linear complexity.* So that the Union-Find routine (given in [2] and recalled in Algorithm 1) remains short, its code does not feature tree balancing; yet it is explained in [3]. *A formal proof of our algorithm.* This paper focuses on how the proposed algorithm works and gives an insight into the reasons why it works; to give a formal proof requires a large amount of materials, the first part of which can be found in [17]. *About high bit-depths data.* That case is not detailled in this paper.

Algorithm 1. "Union-Find"-based computation of a morphological tree.

```
UNION_FIND($\mathcal{R}$) : T ;
begin
    for all p do
        $zpar(p) \leftarrow$ undef
    for $i \leftarrow N - 1$ to 0 do
        $p \leftarrow \mathcal{R}[i]$;
        $parent(p) \leftarrow p$;
        $zpar(p) \leftarrow p$;
        for all
        $n \in \mathcal{N}(p)$ such as $zpar(n) \neq$ undef
        do
            $r \leftarrow$ FIND_ROOT($zpar, n$);
            if $r \neq p$ then
                $parent(r) \leftarrow p$;
                $zpar(r) \leftarrow p$;
    return $parent$
```

```
FIND_ROOT($zpar, x$) : P ;
begin
    if $zpar$ (x) = x then
        return $x$
    else
        $zpar(x) \leftarrow$
        FIND_ROOT($zpar$, $zpar(x)$);
        return $zpar(x)$

COMPUTE_TREE ($u$) : Pair(Array[P], T) ;
begin
    $\mathcal{R} \leftarrow$ SORT($u$);
    $parent \leftarrow$ UNION_FIND($\mathcal{R}$);
    CANONICALIZE_TREE($u$, $\mathcal{R}$, $parent$);
    return ($\mathcal{R}$, $parent$);
```

named *parent*, and encoded as an nD image (so $parent(p)$ is an nD point). When a node of the max-tree contains several points, we choose its first point (with respect to $\mathcal{R}$) as the representative for this node; that point is called a component "canonical point" or a "level root". Let Γ denote a component corresponding to a node of the max-tree, p_Γ its canonical element, and p_r the root canonical element. The *parent* function that we want to construct should verify the following four properties: **1.** $parent(p_r) = p_r$; **2.** $\forall\, p \neq p_r,\ parent(p)\, \mathcal{R}\, p$; **3.** p is a canonical element iff $p = p_r \vee u(parent(p)) \neq u(p)$; **4.** $\forall\, p,\ p \in \Gamma \Leftrightarrow u(p) = u(p_\Gamma) \wedge \exists\, i, parent^i(p) = p_\Gamma$ (therefore $\forall\, p \in \Gamma,\ p = p_\Gamma \vee p_\Gamma\, \mathcal{R}\, p$).

The routine UNION_FIND, given in Algorithm 1, is the classical "union-find" algorithm [19] but *modified* so that it computes the expected morphological tree [2] while browsing pixels following $\mathcal{R}^{-1}$, i.e., from leaves to root (let us recall that we do not feature here the union-by-rank version). Its result is a *parent* function that fulfills those first four properties. Obtaining the following extra property, "5. $\forall p$, $parent(p)$ is a canonical element," is extremely interesting since it ensures that the parent function, when restricted to canonical elements only, gives a "compact" morphological tree such as the ones depicted in Figure 1. Precisely it allows to browse components while discarding their contents: a traversal is thus limited to one element (one pixel) per component, instead of passing through every image elements (pixels). Transforming the parent function so that property 5 is verified can be performed by a simple post-processing of the union-find computation. The resulting tree has now the simplest form that we can expect; furthermore we have an isomorphism between images and their canonical representations.

2.2 Computing the Max-tree and the Tree of Shapes

The algorithm presented in [2] to compute the max-tree is surprisingly also able to compute the tree of shapes. The skeleton, or *canvas*, of this algorithm is the routine COMPUTE_TREE given in the right part of Algorithm 1; it is composed of *three* steps: sort the image elements (pixels); then run the modified union-find algorithm to compute a tree encoded by a parent function; last modify the parent function to give that tree its canonical form.

In the case of the max-tree, the sorting step provides $\mathcal{R}$ encoded as an array of points sorted by increasing gray-levels in u, i.e., such that the array indices satisfy $i < i' \;\Rightarrow\; u(\mathcal{R}[i]) \leq u(\mathcal{R}[i'])$. When image data are low quantized, typically 12 bit data or less, then sorting points can be performed by a distribution sort algorithm. Last, the canonicalization post-processing is a trivial 5-line routine that the reader can find in [2]. In the case of the tree of shapes, it is also a tree that represents an inclusion relationship between connected components of the input image. As a consequence a first important idea to catch is that the tree of shapes can be computed with the exact same routine, UNION_FIND, as the one used by max-tree.

2.3 What If...

The major and crucial difference between the max-tree and the tree of shapes computations is obviously the sorting step. For the UNION_FIND routine to be able to compute the tree of shapes using $\mathcal{R}^{-1}$, the SORT routine has to sort the image elements so that $\mathcal{R}$ corresponds to a downward browsing of the tree of shapes. Schematically we expect that $\mathcal{R}$ contains the image pixels going from the "external" shapes to the "internal" ones (included in the former ones).

The similarity between the computations of both trees is illustrated in Figure 2. We can see that the modified union-find algorithm correctly computes

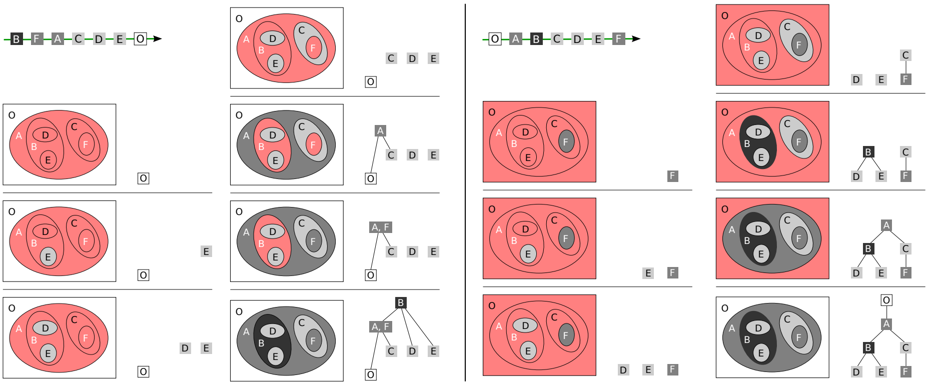
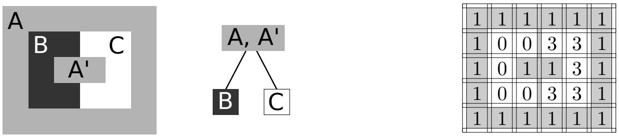


Fig. 2. Tree computation of the max-tree (left) and of the tree of shapes (right). For both cases, the result $\mathcal{R}$ of the sorting step is given over the green arrow and the tree computation, browsing $\mathcal{R}^{-1}$, is progressively depicted.



1	1	1	1	1	1
1	0	0	3	3	1
1	0	1	1	3	1
1	0	0	3	3	1
1	1	1	1	1	1

Fig. 3. A sample image and its tree of shapes (left); a step towards an *ad-hoc* image representation (right)

both trees once $\mathcal{R}$ is properly defined. Therefore we "just" need to know how to compute $\mathcal{R}$ in the case of the tree of shapes to turn the canvas given in the previous Section 2.2 as the expected algorithm.

Let us consider the image depicted on the left of Figure 3 with its tree of shapes. We can see that we need to reach the regions A and A' before the regions B and C in order to properly sort pixels, i.e., to compute $\mathcal{R}$. It is only possible if we can pass "between" pixels. The representation depicted on the right of Figure 3 is well-suited for that since it contains some elements that materialize inter-pixel spaces. Furthermore, given a two adjacent pixels with respective values 0 and 3, the element in-between them has to bear all the "intermediate" values: not only 1 but also 2. Indeed, if we change the value of regions A and A' from 1 to 2, the tree structure is unchanged but inter-pixel elements between regions B and C have now to make A and A' connect with value 2. Eventually we need an image representation that is "continuous" in some way with respect to both the domain space and the value space.

3 Image Representation

To be able to sort the image pixels so that $\mathcal{R}$ corresponds to a top-down browsing of tree of shapes elements, this paper introduces a novel representation of images[4]. It relies on a couple of theoretical tools briefly described hereafter[5].

3.1 Cellular Complex and Khalimsky Grid

From the sets $H_0^1 = \{\{a\}; a \in \mathbb{Z}\}$ and $H_1^1 = \{\{a, a+1\}; a \in \mathbb{Z}\}$, we can define $H^1 = H_0^1 \cup H_1^1$ and the set H^n as the n-ary Cartesian power of H^1. If an element $h \subset \mathbb{Z}^n$ is the Cartesian product of d elements of H_1^1 and $n-d$ elements of H_0^1, we say that h is a d-face of H^n and that d is the dimension of h. The set of all faces, H^n, is called the nD space of cubical complexes. Figure 4 depicts a set of faces $\{f, g, h\} \subset H^2$ where $f = \{0\}\times\{1\}$, $g = \{0,1\}\times\{0,1\}$, and $h = \{1\}\times\{0,1\}$; the dimension of those faces are respectively 0, 2, and 1. Let us write $h^{\uparrow} = \{h' \in H^n \,|\, h \subseteq h'\}$ and $h^{\downarrow} = \{h' \in H^n \,|\, h' \subseteq h\}$. The pair $(H^n, \subseteq)$ forms a poset and the set $\mathcal{U} = \{U \subseteq H^n \,|\, \forall h \in U,\, h^{\uparrow} \subseteq U\}$ is a T0-Alexandroff topology on H^n. With $E \subseteq H^n$, we have a star operator $st(E) = \cup_{h \in E}\, h^{\uparrow}$ and a

[4] In [17], a formal characterization of the discrete topology underlying this novel representation is presented.

[5] The authors recommend [12] and [1] for extra readings about those tools.

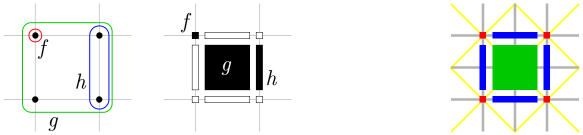


Fig. 4. Three faces depicted as subsets of $\mathbb{Z}^2$ (left) and as geometrical objects (middle); Khalimsky grid (right) with 0- to 2-faces respectively painted in red, blue, and green

closure operator $cl(E) = \cup_{h \in E}\, h^{\downarrow}$, that respectively gives the smallest open set and the smallest closed set of $\mathcal{P}(H^n)$ containing E.

The set of faces of H^n is arranged onto a grid, the so-called Khalimsky's grid, depicted in gray in Figure 4 (right); and inclusion between faces lead to a neighborhood relationship, depicted in gray and yellow. The set of 2-faces, the minimal open sets of H^n, is the n-Cartesian product of H_1 and is denoted by H_1^n.

3.2 Set-Valued Maps

A set-valued map $\textsc{u} : X \rightsquigarrow Y$ is characterized by its graph, $\mathrm{Gra}(\textsc{u}) = \{\, (x,y) \in X \times Y \mid y \in \textsc{u}(x) \,\}$. There are two different ways to define the "inverse" of a subset by a set-valued map: $\textsc{u}^{\oplus}(M) = \{\, x \in X \mid \textsc{u}(x) \cap M \neq \emptyset \,\}$ is the *inverse image* of M by $\textsc{u}$, whereas $\textsc{u}^{\ominus}(M) = \{\, x \in X \mid \textsc{u}(x) \subset M \,\}$ is the *core* of M by $\textsc{u}$. Two distinct continuities are defined on set-valued maps. The one we are interested in is the "natural" extension of the continuity of a single-valued function. When X and Y are metric spaces and when $\textsc{u}(x)$ is compact, $\textsc{u}$ is said to be Upper Semi-Continuous (u.s.c.) at x if $\forall \varepsilon > 0$, $\exists\, \eta > 0$ such that $\forall\, x' \in B_X(x,\eta)$, $\textsc{u}(x') \subset B_Y(\textsc{u}(x), \varepsilon)$, where $B_X(x,\eta)$ denotes the ball of X of radius η centered at x. One characterization of u.s.c. maps is the following: $\textsc{u}$ is u.s.c. if and only if the core of any open subset is open.

3.3 Interpolation

Following the conclusions of Section 2.3, we are going to immerse a discrete nD function defined on a cubical grid $u : \mathbb{Z}^n \to \mathbb{Z}$ into some larger spaces in order to get some continuity properties. For the *domain space*, we use the subdivision $X = \frac{1}{2}H^n$ of H^n. Every element $z \in \mathbb{Z}^n$ is mapped to an element $m(z) \in \frac{1}{2}H_1^n$ with $z = (z_1, \ldots, z_n) \longmapsto m(z) = \{z_1, z_1 + \frac{1}{2}\} \times \ldots \times \{z_n, z_n + \frac{1}{2}\}$. The definition domain of u, $\mathcal{D} \subseteq \mathbb{Z}^n$, has thus a counterpart in X, that will also be denoted $\mathcal{D}$, and that is depicted in bold in Figure 5. For the *value space*, we immerse $\mathbb{Z}$ (the set of pixel values) into the larger space $Y = \frac{1}{2}H^1$, where every integer becomes a closed singleton of H_0^1. Thanks to an "interpolation" function, we can now define from u a set-valued map $\textsc{u} = \mathfrak{I}(u)$. We have $\textsc{u} : X \rightsquigarrow Y$ and we set:

$$\forall\, h \in X,\;\; \textsc{u}(h) = \begin{cases} \{\, u(m^{-1}(h)) \,\} & \text{if } \; h \in \mathcal{D} \\ \max(\, \textsc{u}(h') : \, h' \in st(cl(h)) \cap \mathcal{D}\,) & \text{if } \; h \in \frac{1}{2}H_1^n \backslash \mathcal{D} \\ \mathrm{span}(\, \textsc{u}(h') : \, h' \in st(h) \cap \mathcal{D}\,) & \text{if } \; h \in X \backslash \frac{1}{2}H_1^n. \end{cases} \qquad (1)$$

24	24	24	24	24	24
24	24	0	0	0	24
24	0	6	8	0	24
24	24	0	0	24	24
24	24	24	24	24	24

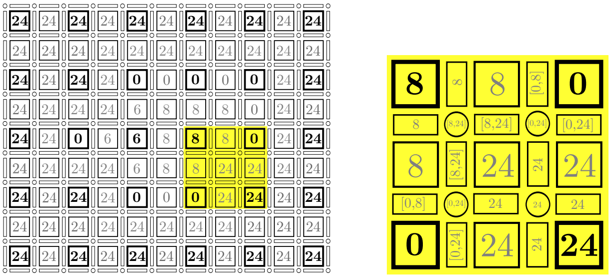


Fig. 5. The function $u : \mathbb{Z}^2 \to \mathbb{Z}$ (left) is transformed into the set-valued map $\text{U} : \frac{1}{2}H^2 \rightsquigarrow \frac{1}{2}H^1$ (middle); d-faces with $d \in \{0, 1\}$ are interval-valued in U with the span of their respective $(d+1)$-face neighbors (right)

An example of interpolation is given in Figure 5. Actually, whatever u, such a discrete interpolation $\mathfrak{I}(u)$ can also be interpreted as a non-discrete set-valued map $\mathfrak{I}_{\mathbb{R}}(u) : \mathbb{R}^n \rightsquigarrow \mathbb{R}$ (schematically $\mathfrak{I}_{\mathbb{R}}(u)(x) = \mathfrak{I}(u)(h)$ with h such as $x \in \mathbb{R}^n$ falls in $h \in \frac{1}{2}H^n$), and we can show that $\mathfrak{I}_{\mathbb{R}}(u)$ is an U.S.C. map.

To the authors knowledge the notion of cuts (or thresholds) have not been defined for set-valued maps. Since they are of prime importance for mathematical morphology, and for the tree of shapes in particular, we propose in this paper the following definitions. Given $\lambda \in Y$, let us state that $[\,\text{U} \lhd \lambda\,] = \{\, x \in X \mid \forall\, \mu \in \text{U}(x),\ \mu < \lambda \,\}$ and $[\,\text{U} \rhd \lambda\,] = \{\, x \in X \mid \forall\, \mu \in \text{U}(x),\ \mu > \lambda \,\}$. We can show [17] that, with those definitions, $\forall u, \forall \lambda$, $[\,\mathfrak{I}(u) \lhd \lambda\,]$ and $[\,\mathfrak{I}(u) \rhd \lambda\,]$ are well-composed [9]. That is, strict cut components and their complementary sets can be handled both with the same unique connectivity, c_{2n}. As a consequence, the operators *star* and Sat commute on those sets, and we can prove [17] that:

$$\mathfrak{S}_{\mathfrak{I}}(u) \;=\; \{\, \text{Sat}_{c_{2n}}(\Gamma);\ \Gamma \in \{\mathcal{CC}_{c_{2n}}([\,\mathfrak{I}(u) \lhd \lambda\,])\}_{\lambda} \cup \{\mathcal{CC}_{c_{2n}}([\,\mathfrak{I}(u) \rhd \lambda\,])\}_{\lambda} \,\}$$

is a set of components that forms a tree. Moreover, we can also prove that $\mathcal{T}_{<}(u) = \{\, \Gamma \cap \mathcal{D};\ \Gamma \in \{\, \mathcal{CC}_{c_{2n}}([\,\mathfrak{I}(u) \lhd \lambda + 1/2\,])\,\}_{\lambda \in H_0}\,\}$ and $\mathcal{T}_{\geq}(u) = \{\, \Gamma \cap \mathcal{D};\ \Gamma \in \{\, \mathcal{CC}_{c_{2n}}([\,\mathfrak{I}(u) \rhd \lambda + 1/2\,])\,\}_{\lambda \in H_0}\,\}$. So eventually we have: $\mathfrak{S}(u) = \{\, \Gamma \cap \mathcal{D};\ \Gamma \in \mathfrak{S}_{\mathfrak{I}}(u) \}$. That final property means that strict cuts of the interpolation of u, considering only c_{2n} for the different operators, allows for retrieving the shapes of u, as defined with the pair of dual connectivities c_{2n} and c_{3^n-1}.

4 Putting Things Altogether

4.1 About Saturation and Initialization

Classically the root node of the tree of shapes represents the whole image and, formally, the saturation operator is defined w.r.t. a point at infinity, p_∞, located outside the image domain $\mathcal{D}$. A rather natural idea is that the root level, ℓ_∞, should only depend on the internal border of $\mathcal{D}$ (which is unfortunately not the case for the algorithms proposed in the literature). To that aim, before interpolating u, we add to this image an external border with a unique value, ℓ_∞, set to the median value of the internal border. p_∞ is then one point from the added border.

4.2 Handling the Hierarchical Queue

To sort the faces of the domain X of U, we use a classical front propagation based on a hierarchical queue [15], denoted by q, the current level being denoted by ℓ. The sorting algorithm is given in Algorithm 2. There are two notable differences with the well-known hierarchical-queue-based propagation. First the d-faces, with $d < n$, are interval-valued so we have to decide at which (single-valued) level to enqueue those elements. The solution is straightforward: a face h is enqueued at the value of the interval $U(h)$ that is the closest to ℓ (see the procedure PRIORITY_PUSH). Just also note that we memorize the enqueuing level of faces thanks to the image $u^\flat$ (see the procedure SORT). Second, when the queue at current level, $q[\ell]$, is empty (and when the hierarchical queue q is not yet empty), we shall decide what the next level to be processed is. We have the choice of taking the next level, either less or greater than ℓ, such that the queue at that level is not empty (see the procedure PRIORITY_POP). Practically choosing going up or down the levels does not change the resulting tree since it just means exploring some sub-tree before some other disjoint sub-tree.

Algorithm 2. Sorting for tree of shapes computation.

```
PRIORITY_PUSH($q$, $h$, $U$, $\ell$);
/* modify $q$ */;
begin
    [$lower$, $upper$] $\leftarrow U(h)$;
    if $lower > \ell$ then
        $\ell' \leftarrow lower$;
    else if $upper < \ell$ then
        $\ell' \leftarrow upper$;
    else
        $\ell' \leftarrow \ell$;
    PUSH($q[\ell'], h$);

PRIORITY_POP(q, $\ell$) : H ;
/* modify $q$, and sometimes $\ell$ */;
begin
    if $q[\ell]$ is empty then
        $\ell' \leftarrow$ level next to $\ell$ such as $q[\ell']$
        is not empty;
        $\ell \leftarrow \ell'$;
    return POP($q[\ell]$);

SORT($U$) : Pair(Array[H], Image) ;
begin
    for all h do
        $deja\_vu(h) \leftarrow$ false;
    $i \leftarrow 0$;
    PUSH($q[\ell_\infty]$, $p_\infty$);
    $deja\_vu(p_\infty) \leftarrow$ true;
    $\ell \leftarrow \ell_\infty$  /* start from root level */ ;
    while $q$ is not empty do
        $h \leftarrow$ PRIORITY_POP($q$, $\ell$);
        $u^\flat(h) \leftarrow \ell$;
        $\mathcal{R}[i] \leftarrow h$;
        for all $n \in$
        $\mathcal{N}(h)$ such as $deja\_vu(n) =$ false
        do
            PRIORITY_PUSH($q$, $n$, $U$, $\ell$);
            $deja\_vu(n) \leftarrow$ true;
        $i \leftarrow i + 1$;
    return ($\mathcal{R}$, $u^\flat$)
```

The result $\mathcal{R}$ of the sorting step is the one expected since the image U, in addition with the browsing of level in the hierarchical queue, allows for a propagation that is "continuous" both in domain space and in level space. An interesting property due to the interpolation and the well-composedness of cuts is that the neighborhood $\mathcal{N}$, used for faces in the propagation, corresponds to the c_{2n} connectivity on the Khalimsky's grid.

4.3 Max-tree versus Tree of Shapes Computation

The main body of the tree of shapes computation algorithm is given in Algorithm 3. The major differences between this algorithm and the one dedicated to the max-tree (see the procedure COMPUTE_TREE in Algorithm 1) are the following ones.

Algorithm 3. Tree of shapes computation in five steps.

```
COMPUTE_TREE_OF_SHAPES ($u$) : Pair(Array[P], T) ;
begin
    $U \leftarrow$ INTERPOLATE($u$);
    ($\mathcal{R}$, $u^\flat$) $\leftarrow$ SORT($U$);
    parent $\leftarrow$ UNION_FIND($\mathcal{R}$);
    CANONICALIZE_TREE($u^\flat$, $\mathcal{R}$, parent);
    return UN-INTERPOLATE($\mathcal{R}$, parent);
```

First the three basic steps (sort, union-find, and canonicalization) are now surrounded by an interpolation and un-interpolation process. Note that the uninterpolation just cleans up both $\mathcal{R}$ and *parent* to keep only elements of $\mathcal{D}$. Second, as emphasized in Section 2.2, the sorting step is of course dedicated to the tree of shapes computation. Last, a temporary image, $u^\flat$, is introduced. It is defined on the same domain as U, namely X, and contains only single-valued elements. This image is the equivalent of the original image u when dealing with the max-tree: it is used to know when an element h is canonical, that is, when $u^\flat(parent(h)) \neq u^\flat(h)$ (so that image is thus required by the canonicalization step that runs on X).

Complexity analysis of the algorithm presented here is trivial. The interpolation, canonicalization, and un-interpolation are linear. The modified union-find (once augmented with tree balancing, i.e., union-by-rank) is quasi-linear when values of the input image u have a low quantization (typically 12 bits or less). Last, the time complexity of the sorting step is governed by the use hierarchical queue: it is linear with low quantized data[6]. Eventually we obtain a quasi-linear algorithm. The representation of the tree with the pair ($\mathcal{R}$, *parent*) allows for any manipulation and processing that one expects from a morphological tree [3].

[6] Formally the sorting step has the pseudo-polynomial $O(k\,n)$ complexity, k being the number of different gray values. Though, since we consider low bit-depths data, k shall only be considered as a complexity multiplicative factor.

5 Related Works

The first known algorithm, the "Fast Level Line Transform (FLLT)" [16], computes the max-tree and the min-tree of an image and obtains the tree of shapes by merging both trees. The main drawback of the FLLT is the need to know that a component has an hole (in order to match it with a component of the other tree). To that aim the Euler characteristic is computed, which can be done *locally* (while following the border of components) but in 2D only. In [4,14] the authors show that this fusion approach is sound in nD with $n > 2$; yet it cannot be effective in practice due to unacceptable complexity.

In [5] the "Fast Level Set Transform" (FLST) relies on a region-growing approach to decompose the image into shapes. It extracts each branch of the tree starting from the leaves and growing them up to the root until at least one saddle point is encountered. Each time a saddle point is encountered, the branch extraction procedure has to stop until every parallel branch meeting at this point is extracted. So each saddle point invalidates the shape currently being extracted, forcing the algorithm to visit its pixels again once a parallel branch is extracted. Since an image like a checkerboard contains $\mathrm{O}(n)$ saddle points meeting on $\mathrm{O}(n)$ pixels, the FLST has a $\mathrm{O}(n^2)$ worst case time complexity.

Song [18] takes a top-down approach to build the *tree of level lines* in $\mathrm{O}(n+t)$ time, where t is the total length of all level lines (note that filling the interior of each level line allows for retrieving the tree of shapes). The algorithm is restricted to 2D images with hexagonal pixels. Its key idea is to perform a recursion (starting from the image boundary): for a given component, follow every contours of its holes, and repeat this procedure for each hole component. Since the total length of level lines of an image can be of order $\mathrm{O}(n^2)$, the worst case has a quadratic-time complexity.

6 Conclusion

In this paper, we have presented a new algorithm to compute the tree of shapes of an image which features a quasi-linear time complexity, runs on nD images, and benefits from a much simpler implementation than existing algorithms. We have also proposed a novel representation of images as set-valued maps which has some continuity properties while remaining discrete.

Actually we believe that this representation is a good start to get a *"pure" self-duality* for images and operators, that is, a way to get rid of the pair of dual connectivities c_{2n} and c_{3^n-1}, and of the dissymmetry of cuts (strict and large cuts for respectively lower and upper cuts). In particular, replacing the maximum operator by the median operator in Equation 1 leads to a pure self-dual definition of the tree of shapes of 2D images [17]. Furthermore the perspectives offered by that new representation might be far from being limited to the tree of shapes computation.

For our experiments we use our free software library [10]; in particular, the fact that our tool makes it easy to write generic software in the case of mathematical morphology and discrete topology is discussed in [11]. The work presented here will be available in the next release of our software for we advocate reproducible research.

Acknowledgements. The authors would like to thanks Michel Couprie and Jean Cousty for fruitful discussions. This work received funding from the Agence Nationale de la Recherche, contract ANR-2010-BLAN-0205-03 and through "Programme d'Investissements d'Avenir" (LabEx BEZOUT n°ANR-10-LABX-58).

References

1. Aubin, J.P., Frankowska, H.: Set-Valued Analysis. Modern Birkhäuser Classics, Birkhäuser (2008)
2. 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: Proceedings of ICIP, vol. 4, pp. 41–44 (2007)
3. Carlinet, E., Géraud, T.: A comparison of many max-tree computation algorithms. In: Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 73–95. Springer, Heidelberg (2013)
4. Caselles, V., Meinhardt, E., Monasse, P.: Constructing the tree of shapes of an image by fusion of the trees of connected components of upper and lower level sets. Positivity 12(1), 55–73 (2008)
5. Caselles, V., Monasse, P.: Geometric Description of Images as Topographic Maps. Lecture Notes in Mathematics Series, vol. 1984. Springer (2009)
6. Géraud, T.: Ruminations on tarjan's union-find algorithm and connected operators. In: Proceedings of ISMM. CIVS, vol. 30, pp. 105–116. Springer (2005)
7. Géraud, T., Talbot, H., Van Droogenbroeck, M.: Mathematical Morphology—From Theory to Applications, ch. 12, pp. 323–353. ISTE & Wiley (2010)
8. Henle, M.: A Combinatorial Introduction to Topology. Dover Publications Inc. (1994)
9. Latecki, L., Eckhardt, U., Rosenfeld, A.: Well-composed sets. Computer Vision and Image Understanding 61, 70–83 (1995)
10. Levillain, R., Géraud, T., Najman, L.: Why and how to design a generic and efficient image processing framework: The case of the Milena library. In: Proceedings of ICIP, pp. 1941–1944 (2010), http://olena.lrde.epita.fr
11. Levillain, R., Géraud, T., Najman, L.: Writing reusable digital topology algorithms in a generic image processing framework. In: Köthe, U., Montanvert, A., Soille, P. (eds.) WADGMM 2010. LNCS, vol. 7346, pp. 140–153. Springer, Heidelberg (2012)
12. Mazo, L., Passat, N., Couprie, M., Ronse, C.: Digital imaging: A unified topological framework. Journal of Mathematical Imaging and Vision 44(1), 19–37 (2012)
13. Meijster, A., Wilkinson, M.H.F.: A comparison of algorithms for connected set openings and closings. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(4), 484–494 (2002)
14. Meinhardt-Llopis, E.: Morphological and Statistical Techniques for the Analysis of 3D Images. Ph.D. thesis, Universitat Pompeu Fabra, Spain (March 2011)
15. Meyer, F.: Un algorithme optimal de ligne de partage des eaux. In: Actes du 8e congrès AFCET, pp. 847–859 (1991)

16. Monasse, P., Guichard, F.: Fast computation of a contrast invariant image representation. IEEE Transactions on Image Processing 9(5), 860–872 (2000)
17. Najman, L., Géraud, T.: Discrete set-valued continuity and interpolation. In: Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 37–48. Springer, Heidelberg (2013)
18. Song, Y.: A topdown algorithm for computation of level line trees. IEEE Transactions on Image Processing 16(8), 2107–2116 (2007)
19. Tarjan, R.E.: Efficiency of a good but not linear set union algorithm. Journal of the ACM 22(2), 215–225 (1975)
20. Xu, Y., Géraud, T., Najman, L.: Context-based energy estimator: Application to object segmentation on the tree of shapes. In: Proceedings of ICIP (2012)
21. Xu, Y., Géraud, T., Najman, L.: Morphological filtering in shape spaces: Applications using tree-based image representations. In: Proceedings of ICPR (2012)
22. Xu, Y., Géraud, T., Najman, L.: Two applications of shape-based morphology: Blood vessel segmentation and generalisation of constrained connectivity. In: Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 390–401. Springer, Heidelberg (2013)

Efficient Schemes for Computing α-tree Representations

Jiří Havel[1], François Merciol[2], and Sébastien Lefèvre[2]

[1] Brno University of Technology, Czech Republic
ihavel@fit.vutbr.cz
[2] Université de Bretagne-Sud, IRISA, France
{francois.merciol,sebastien.lefevre}@univ-ubs.fr

Abstract. Hierarchical image representations have been addressed by various models by the past, the max-tree being probably its best representative within the scope of Mathematical Morphology. However, the max-tree model requires to impose an ordering relation between pixels, from the lowest values (root) to the highest (leaves). Recently, the α-tree model has been introduced to avoid such an ordering. Indeed, it relies on image quasi-flat zones, and as such focuses on local dissimilarities. It has led to successful attempts in remote sensing and video segmentation. In this paper, we deal with the problem of α-tree computation, and propose several efficient schemes which help to ensure real-time (or near-real time) morphological image processing.

Keywords: α-tree, Quasi-Flat Zones, Image Partition, Hierarchies, Efficient Algorithms.

1 Introduction

Mathematical morphology has long been a provider of interesting hierarchical image representations, mainly by trees, e.g. component tree [4], min and max-tree [9], binary partition tree [8], etc. The max-tree (and its respective counterpart, min-tree) has been widely used due to its nice properties as well as the availability of efficient algorithms to first compute the tree from an image, and second process the tree (e.g. with a filtering to remove irrelevant nodes), thus leading to the processing of the underlying image.

Recently, a new image model, namely the α-tree [6], has been introduced to avoid relying on an ordering relation among image pixels. This model is a hierarchical representation of the quasi-flat zones of an image, and as such, relies on local dissimilarities α. While this model already led to successful attempts in exploration of remote sensing data [5] and image/video segmentation [3], it still requires some efficient computing schemes in order to be widely disseminated and to address a large scope of problems. In this paper, we focus on computational issues of the α-tree model, and propose new efficient algorithms to build this α-tree.

The rest of this paper is organized as follows. Section 2 provides necessary background on quasi-flat zones and α-tree. Various schemes for computing efficiently the α-tree are discussed in Section 3. Section 4 is devoted to the presentation of experimental results, while concluding remarks are given in Section 5.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 111–122, 2013.

2 Background

The α-tree image model is a multiscale representation of an image through its α-zones. We recall here the notions of flat zones, quasi-flat zones (including α-zones) and finally the recent α-tree model.

In the following, we will use the notations used in [6]. We will denote by I a digital image and E its definition domain. Let us recall that an image segmentation is a partition $\mathbf{P}$ of E, i.e. a mapping $x \to \mathbf{P}(x)$ from E into $\mathcal{P}(E)$ such that $\forall x \in E \Rightarrow x \in \mathbf{P}(x)$ and $\forall x, y \in E \Rightarrow \mathbf{P}(x) = \mathbf{P}(y)$ or $\mathbf{P}(x) \cap \mathbf{P}(y) = \emptyset$, with $\mathbf{P}(x)$ indicating a cell of $\mathbf{P}$ containing a point $x \in E$. We thus have $\bigcup_{x \in E} \mathbf{P}(x) = E$. Moreover, we will write $\pi(x \rightsquigarrow y)$ a path of length N between any two elements $x, y \in E$, i.e. a chain of pairwise adjacent elements $\langle x = x_0, x_1, \ldots, x_{N-1} = y \rangle$. Finally, let $\Pi \neq \emptyset$ be the set of all possible paths between x and y. The minimum dissimilarity metric between x and y is defined as

$$\hat{d}(x, y) = \bigwedge_{\pi \in \Pi} \left\{ \bigvee_{i \in [0, \ldots, N-1]} \{ d(x_i, x_{i+1}) \mid x_i, x_{i+1} \in \Pi \} \right\} \tag{1}$$

with $d(x, y)$ a predefined dissimilarity measure between attributes of x and y (i.e. pixel intensities).

In a digital image, flat zones are defined as connected sets of pixels sharing the same value. Formally, the flat zone of x is defined as

$$\begin{aligned} \mathcal{Z}(x) = \{x\} \ \cup \ \{y \mid \exists \pi(x \rightsquigarrow y) :& \\ & \forall x_i \in \pi(x \rightsquigarrow y) \ \wedge \ x_i \neq y \Rightarrow d(x_i, x_{i+1}) = 0\}. \end{aligned} \tag{2}$$

In the field of Mathematical Morphology, flat zones have been shown to be elements with nice properties [10]. Indeed, the partition of an image into its flat zones most often includes any relevant image segmentation, since objects edges are located between neighboring pixels with different values, i.e. belonging to different flat zones. However, the practical usage of flat zones is limited since it leads to an extreme oversegmentation, flat zones being made of only a few pixels. To counter this problem, softer definitions have been introduced under the name quasi-flat zones. A recent survey related to quasi-flat zones is provided by Soille in [11].

The simplest and most widely used definition of quasi-flat zones is called α-zone. For a given pixel x, its α-zone noted α-$\mathcal{Z}(x)$ is made of all pixels reachable from x through a path with intermediary steps not higher than α. Using the previous definitions, we have

$$\begin{aligned} \alpha\text{-}\mathcal{Z}(x) = \{x\} \ \cup \ \{y \mid \exists \pi(x \rightsquigarrow y) :& \\ & \forall x_i \in \pi(x \rightsquigarrow y) \ \wedge \ x_i \neq y \Rightarrow d(x_i, x_{i+1}) \leq \alpha\}, \end{aligned} \tag{3}$$

the specific case of $\alpha = 0$ leading to standard flat zones. Let us observe that α-zones define a partition or segmentation, i.e. $\bigcup_{x \in E} \alpha\text{-}\mathcal{Z}(x) = E$ and $\forall x, y \in E$: $\alpha\text{-}\mathcal{Z}(x) \cap \alpha\text{-}\mathcal{Z}(y) \neq \emptyset \implies \alpha\text{-}\mathcal{Z}(x) = \alpha\text{-}\mathcal{Z}(y)$. The main drawback of α-zones is their purely local behavior, which can lead to an artifact called chaining effect. This is observed when the successive steps in a path $\pi(x \rightsquigarrow y)$ are low (w.r.t α) while the dissimilarity measure between x and y is high, for instance in the case of a gradual transition from black to white. This limitation was addressed by Soille [11], and later Soille and Grazzini [12], who added another constraint that prevented the uncontrolled α-zone from growing.

Another way to use the α-zones has been recently reported by Ouzounis and Soille in [6]. In this seminal work, they introduce the concept of α-tree based on a partition pyramid. Since the α zones can be ordered by inclusion relation, it is possible to construct a tree of α-zones. The root of the tree is a zone covering the whole image. Every parent node contains a zone, that is a superset of zones contained in its children.

Every level of the α-tree contains all zones for a specific value of α. The α-tree contains all possible image segmentations based on α-zones. Every cut through the α-tree selects an image segmentation. To benefit from this powerful image representation, efficient computation schemes are required.

3 Computing the α-tree

3.1 Basic Principles

In the previous section, we have introduce the α-tree reusing its standard notations [6]. Let us observe that the α-tree can also be defined through graphs. We will use this latter representation to introduce efficient computation scheme. Thus, let us also denote an image by a graph $(\mathcal{V}, \mathcal{E}, \mathcal{I})$, where $\mathcal{V}$ are the image pixels, $\mathcal{E}$ are the edges between them and $\mathcal{I} \subseteq \mathcal{V} \times \mathcal{E}$ is the incidence relation between vertices and edges, i.e. for edge e, that connects v_1 and v_2, $v_1\mathcal{I}e$ and $v_2\mathcal{I}e$. The α is a weight of the edges of the graph.

The α-tree can be constructed as a min-tree built over an edge graph, as outlined by Soille and Najman in [13]. An edge graph for a graph $(\mathcal{V}, \mathcal{E}, \mathcal{I})$ is a graph $(\mathcal{V}', \mathcal{E}', \mathcal{I}')$, with edges and vertices exchanged $(\mathcal{V}' = \mathcal{E}, \mathcal{E}' = \mathcal{V})$, while the incidence is preserved $(e\mathcal{I}'v = v\mathcal{I}e)$. Using this definition, the edge graph is actually a multigraph, i.e., for a 4-connected image the edges connect up to four vertices. The min-tree of the edge graph can be then transformed in an α-tree. As Soille and Najman mention in [13], this indirect construction is unnecessary, but their paper does not provide a direct algorithm.

It is possible to build the α-tree directly by a modification of Tarjan's union-find algorithm [14]. Tarjan's algorithm was initially defined to identify connected components, it has been later used for construction of different component trees, e.g. in [4]. The union-find method processes a list of graph edges sorted by increasing alpha. For each edge it finds two largest connected components, the edge connects, and merges them. Since the algorithm works with sorted edges, the α-tree is built bottom-up from fine to coarse subdivision of the image.

An edge of the image graph connects two neighboring pixels. Algorithm 1 shows the basic α-tree construction. The parameters of an edge are its two endpoints and the corresponding α. The tree is built bottom-up by subsequent merges of partial trees. For every edge, the two partial trees that contain the edge endpoints are found and then merged. When a pixel is not yet linked to a leaf node, the node is created and linked from the leaf array. Due to the ordered edges, the nodes will be merged only in the active layer. It is the layer of the tree with α being equal to the α of the processed edge. The levels of the tree below the active level are never changed again.

To ensure, that each path from a root to a leaf contains at most one node per α-level, there are four possibilities of merging the subtrees. If both tree roots have α less than the processed edge, a new node with both trees as children is created. If only one tree root has α less than the processed edge, then this tree is attached as a child to the other tree root. Finally if both roots have α equal to the processed edges, then the two root nodes are merged (i.e. their children attached to one of them and the other marked for removal)

Algorithm 1. Algorithm for α-tree construction

```
Require: Image $I$
Ensure: Array leaves[height($I$), width($I$)] of Node pointers
  leaves$[i, j]$ := null, $\forall i, j$
  $E$ = edges($I$)
  sort $E$ by ascending $\alpha$
  for $e = e_1 \to e_{|E|}$ do
      $p_1, p_2$ = points($e$)
      $n_{\{1,2\}}$ = findRoot($p_{\{1,2\}}$)
      if $n_i$ = null then
          leaves$[p_i] = n_i$ = makeNode($p_i, \alpha(e)$)
      end if
      if $n_1 \neq n_2$ then
          if $\alpha(n_1) = \alpha(n_2) = \alpha(e)$ then
              merge($n_1, n_2$)
          else if $\alpha(n_1) < \alpha(e)$ and $\alpha(n_2) < \alpha(e)$ then
              makeNode($n_1, n_2, \alpha(e)$)
          else if $\alpha(n_1) = \alpha(e)$ and $\alpha(n_2) < \alpha(e)$ then
              attach $n_2$ to $n_1$
          else if $\alpha(n_1) < \alpha(e)$ and $\alpha(n_2) = \alpha(e)$ then
              attach $n_1$ to $n_2$
          end if
      end if
      Update path compression cache
  end for
```

When the α is a small integer, the edges can be ordered in $O(E)$ time using the bucketsort algorithm. This algorithm first calculates a histogram of edge alphas. Calculating the prefix sum converts the histogram to a list of offsets to an output array. The reordering is then done in the following pass.

Finding the root of a subtree for a given pixel contains potentially lengthy bottom-up tree traversal. The traversal can be optimized by path compression. During the tree construction, a cache exists, that links tree nodes to the subtree roots. After each processed edge, this cache is updated. Because the number of affected pixels is large and a full update would be costly, only portion of the affected nodes is updated. Thus, the cache does not remove the bottom-up tree traversal, but can significantly shorten it. On the other hand, nodes can't be deleted during the merge process, because the cache can still link to them. It is necessary to retain a list of such nodes and delete them after the cache is no longer needed.

As this algorithm is based on union-find, the time complexity is pseudolinear if bucketsort can be used or $O(N \log N)$ (N is the edge count, in case of the images also the pixel count) otherwise. However there is one important limitation. The node merge must be done in constant (or potentially logarithmic) time, with a (pseudo)linear (or linearithmic) postprocessing step.

Compared to other connected component tree algorithms based on union-find, this original algorithm guarantees that each path from a tree leaf to the root contains at most one node for each possible value of α. This is important for distance functions such as the one used by [3] where the α is discrete with known upper bound. This reduces the traversal cost to an amortized constant time, independent of the tree size. This speeds up the tree construction at the expense of more complicated tree structure with larger memory footprint.

3.2 C++ Implementation

The common implementation of a component tree is an array of parent indices. This representation is very memory efficient, as the tree requires only $4wh$ bytes of memory. However this representation allows easily only bottom-up tree traversal. Here we used more complex representation for tree nodes.

Every node of the tree consists of a link to a parent node, a set of links to children and a set of leaf pixels. Additionally, the node contains the α value of the edge that created it. The parent pointer of a root node is NULL. The sets of children and pixels allow for very simple iteration over the pixels of a connected component. The set of all pixels of a subtree is a recursive union of pixels of a node and all of its children. The sets are represented by linked lists:

```
struct Node
{
    uint8_t alpha;
    Node * parent;
    Node * first_child, * last_child,
    Node * next_sibling, * prev_sibling;
    Leaf * first_leaf, * last_leaf;
};
```

Children and leaves are represented by doubly linked lists (node removal and list merge have $O(1)$ complexity).

The leaf structure does not contain any data yet. All leaves are stored in an array and the position in the array encodes the pixel position.

```
struct Leaf
{
    Node * parent;
    Leaf * next_sibling, * prev_sibling;
};
```

Compared to an array of parent indices, this representation requires more complicated merging of tree nodes, but the resulting tree is as flat as possible so the tree traversal is shorter. Also, tree balancing is not necessary. Figure 1 shows a sample α-tree and the related linked structure.

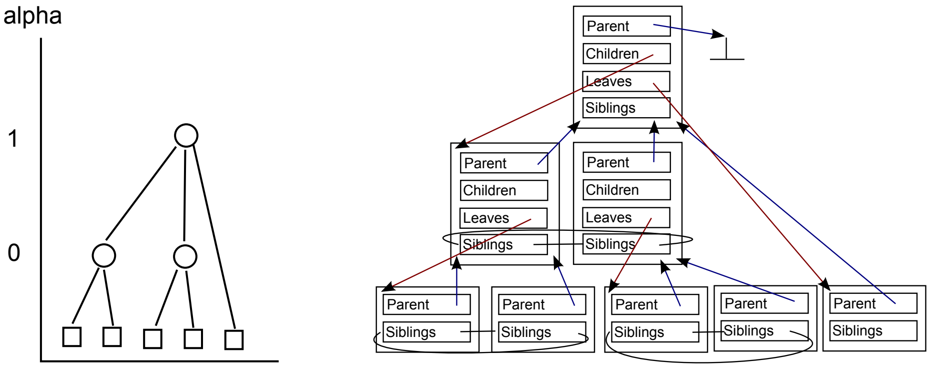


Fig. 1. Sample tree shape and corresponding simplified set of Nodes, Leaves and connecting pointers (for the sake of clarity, only a subset of pointers has been shown)

The maximal height of the tree depends on the datatype of α. In the case of discrete α it is at most the number of α levels that appear in the input image (i.e., at most 256 in case of uint8_t). If α is continuous, it is at most wh when the tree degrades to a simple list. The tree depth of this tree will be always less or equal to the depth of a binary tree implemented by an array of parent links. The cost is lower cache coherence because of dynamic allocation and larger node size.

Two practical strategies for path compression cache are possible:

- Root is cached per node. Cache of nodes visited during tree traversal get updated. This strategy is closer to original Tarjan's paper [14]. However every node needs another pointer that is unused after tree construction. When the node size is not a limiting factor, then this approach has the best results.
- Root is cached per leaf. Only the edge endpoints are updated. Each vertex is usually visited by four edges, so it is a viable strategy, when the node size should be kept as low as possible.

Because merging of two nodes includes transfer (and reconnection) of child nodes and leaf pixels, it has $O(N)$ complexity. Therefore the actual merge must be deferred to the postprocessing step at the end of the algorithm. Since the removal of the nodes must be done in the postprocessing step anyway, it does not introduce new data structure or steps. The node members for the linked list of children are used for a linked list of nodes marked for removal.

3.3 Extension to Multi-threading

Although the previous algorithm is effective, it is inherently single threaded. The only easily parallelizable part is the extraction of edges and their ordering. The tree levels must be processed sequentially. Even in one level, the parallelization would require synchronization of a large portion of the algorithm.

It is however possible to subdivide the input image into multiple parts, build partial trees for them and then merge the resulting trees by a more complicated algorithm that does not need the edges ordered. When the input image is subdivided into regions as square as possible, the number of edges between the subtrees is very low compared to the amount of edges for each subtree. This is the parallelization strategy for other component trees too [2].

The connecting edges are processed sequentially. For both pixels of the connecting edge, the path from the leaf to the tree root is traversed and nodes are found. Contrary to the previous algorithm (where only a subtree root was searched), two tree nodes are required. First node has α less or equal to the connecting edge and the other has α greater than the connecting edge. First nodes are detached and merged in the same way as described by Algorithm 1. Second nodes start two paths to the tree root that need to be merged in a zipper way.

First has α less or equal to the alpha of the connecting edge. These two nodes will be merged in the same way as in the Algorithm 2. The other nodes are the direct parents of the previous nodes, so they have α greater than the connecting edge. These nodes start two paths to tree roots, that will be merged in a way similar to a zipper. This is shown by Algorithm 2.

Algorithm 2 starts by creation of a common root node. This greatly simplifies the zipping process, since the first zip merges the tree roots and subsequent zips merge only inner paths of the tree. The common root is created by merging of roots of T_1 and T_2 when their α is equal or by attaching one to the other. In an unlike case, when all connecting edges have α greater than the roots, a new root node is created and no zipping is done.

4 Experiments

We present here a set of experiments aiming to assert the performance of the proposed computation schemes for building the α-tree. We first present the dataset we are relying on and how we deal with color images, before addressing the respective cases of single and multi-threading. After comparing with [3], this section ends with a discussion about the possibility of processing only partially the tree to lower the computational cost.

4.1 Dataset

We have measured the performance of the proposed algorithms on a few different image datasets. The first is made of the test images of Berkeley segmentation dataset [1] (300 images of size 481 × 321 pixels). The second dataset has been manually built from 8 wallpaper images of size 1920 × 1080 pixels (see Figure 2 for a visual composition).

Algorithm 2. Algorithm for merging of two partial α-trees

```
Require: Trees $T_1, T_2$; Connecting edges $E$
Ensure: Merged tree $T$
  Create common root node.
  for all $e \in E$ do
    $p_1, p_2 = \mathrm{points}(e)$
    $n_{\{}1, 2\} = \mathrm{findNode}(p_{\{}1, 2\}, \leq \alpha(e))$
    $a = \mathrm{findNode}(p_1, > \alpha(e))$
    $b = \mathrm{findNode}(p_2, > \alpha(e))$
    Detach $n_1, n_2$ from $a, b$
    $n = \mathrm{merge}(n_1, n_2)$
    if $\alpha(a) > \alpha(b)$ then
      $\mathrm{swap}(a, b)$
    end if
    Attach $n$ to $a$
    while $a \neq b$ do
      if $\alpha(a) = \alpha(b)$ then
        $n = \mathrm{merge}(a, b)$
        $a = \mathrm{parent}(a)$
        $b = \mathrm{parent}(b)$
      else
        $a = \mathrm{parent}(a)$
      end if
      if $\alpha(a) > \alpha(b)$ then
        Detach $n$ from $a$
        $\mathrm{swap}(a, b)$
        Attach $n$ to $a$
      end if
    end while
  end for
```

Since the segmentation dataset images were too small for a reliable time measurement, we only present here the results obtained with the second dataset. For extra large images, we also used a selection of aerial and satellite photos from mapart.com, namely the 10m SPOT image, Pictometry and DigitalGlobe photos.

Let us mention that images considered here are RGB images, while the definition of the α-tree has been provided only for grayscale images. We have used here the method proposed in [3] to apply the α-tree on RGB images, and we measure the local dissimilarity between neighboring pixels with the Chebyshev distance. Let us consider two colors **c** $= (r, g, b)$ and **c'** $= (r', g', b')$, the Chebyshev distance between **c** and **c**$'$ is given by $\max(|r - r'|, |g - g'|, |b - b'|)$. Since the Chebyshev distance is the L_∞ norm of the absolute difference vector $|c - c'|$, the range of possible distance values is kept low and similar to the input range of each color component (i.e. 256 for an 8-bit image). On the opposite, Manhattan or (worst) Euclidean distance lead to a range respectively equal to 3 × 256 and 256^3 possible distance values. Since this range is to be compared with the various α values, we prefer to set the depth of the α-tree using $A = 256$ rather than $A = 16,777,216$ for practical reasons related to computation time. Let us note,

Fig. 2. A visual composition of the wallpaper dataset showing the 8 wallpaper images used in our experiments

however, that further options might be explored and that extension of α-tree to color and multivariate images will most probably be a direction of future work.

4.2 Single-Threading

We first evaluate the performance of α-tree computation using a single-thread strategy. When the α-tree is built, the average build time is 32ms (15-47ms) for the segmentation dataset and 0.63s (0.44-0.8s) for the wallpapers. According to the program profiling, almost 50% of the build time is spent on bottom-up tree traversal and on updates of the links for path compression. These are the inner loops of the algorithm. All other parts of the algorithm are significantly less time-consuming. The extraction and sorting of edges takes only 10% of the time.

The detailed graphs in Figure 3 shows the detailed timings, number of nodes and rough memory requirements. The construction time increases approximately linearly with the increasing number of the nodes of the tree. The linear dependency is even more significant for larger images as shown on the right graph of this figure. The timing for smaller images is more noisy and contain an outlier probably because of differences in tree structure that turn unimportant with larger input sizes.

4.3 Multi-threading

Performance of parallel tree construction process was measured on a Core i7 2670 CPU with 8 GB of RAM. This processor contains four hyperthreading cores, so it has eight virtual cores. The Figure 4 shows the performance of the parallel construction using multiple threads. The tree construction is mostly memory intensive, so the performance is limited by the memory throughput and not by the raw computation power.

Since the L3 cache and the memory interface is shared between all four cores on Core i7, the memory intensive tree traversal could cause the scaling to stop at a different value than the number of physical or virtual cores. We suspect, that the performance drop for five and more threads is caused by saturation of the L2 cache throughput.

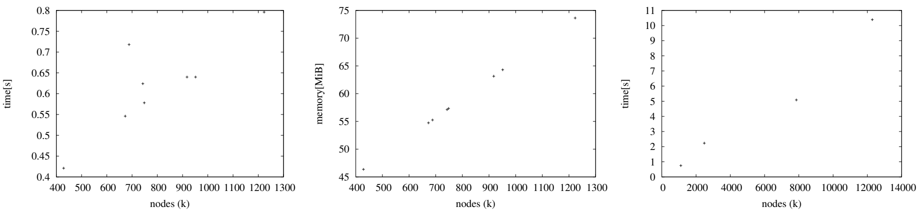


Fig. 3. Dependency of tree construction time [s] (left) and the memory consumption [MiB] (middle) on the number of nodes of the resulting tree, and of tree construction time [s] for large sized images (right)

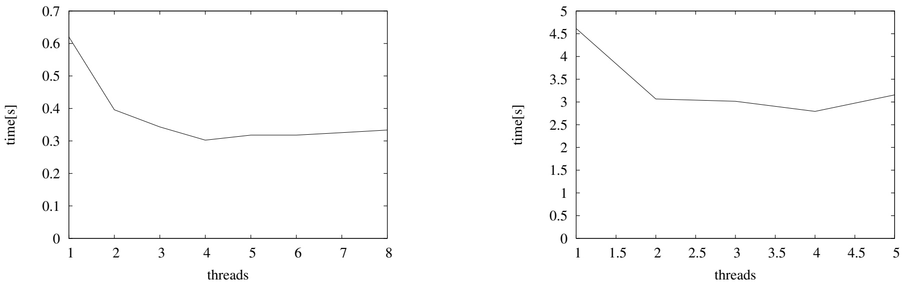


Fig. 4. The average tree build time for increasing thread number. Left: Wallpaper dataset. Right: Satellite+Aerial dataset.

4.4 Comparison with Related Work

In order to assert the efficiency of our algorithm, we compare it with the only available implementation, i.e. [3] which does not rely on the union-find strategy. While our approach requires to collect edges and compress paths, the solution proposed in [3] relies on an insertion process with late merging.

For the sake of comparison, both algorithms have been implemented in Java with the same data structure and compared using two different hardware environments, respectively based on a standard laptop and on a high-performance cluster. This processing environment raises some artifacts (and in particular the influence of the Java garbage collector process) which have been lowered when relevant by computing average times from 10 executions per image (using a trimmed mean to avoid outliers). The processing times given in Table 1 measure the time required to compute the α-tree on the Berkeley Segmentation Dataset (the average time over 300 images is provided), a HD wallpaper image (1920×1080 pixels), and a larger satellite image ($>$ 20 Mpixels). In this context, the focus has to be put rather on the relative times of one algorithm w.r.t. the others than on the absolute time measurement.

From the results given in Table 1, we can see that, while [3] (line 2) improves computation time over the standard union-find strategy (line 1) especially with large images. In order to achieve a fairer comparison, we have developed several extensions of the method proposed [3]. First, we have combined its advantages with the ones offered by the union-find strategy, leading to better results (line 3) on very large images. Second, we have designed a multithreaded version of [3] to be compared with the multithreaded version of our algorithm presented in this paper. We can see that this multithreaded extension of [3] (line 4) naturally better exploits the availability of a parallel machine w.r.t its singlethreaded counterpart. Finally, our method achieves the best results on small images for its singlethreaded version (line 5) which is penalized by the garbage collector process on large images (not observed in the C++ version), and performs best on all images and environments for its multithreaded version (line 6).

Let us observe however that sometimes the algorithm from [3] and its extensions may lead to relatively close results to ours. This comes from the fact that these two approaches are complementary since their optimizations occur at different steps. This will probably motivate for designing a joint algorithm to achieve further optimization.

Table 1. CPU time comparison with [3] for the different steps of α-tree computation (in milliseconds): (1) average on 300 Berkeley Images (154 401 pixels) with Intel Core Duo T9600 @ 2.80GHz; (2) average on 300 Berkeley Images (154 401 pixels) with 12 Intel Xeon L5640 @ 2.27GHz; (3) Wallpaper HD (2 073 600 pixels) with 12 Intel Xeon L5640 @ 2.27GHz; (4) satellite image (23 403 033 pixels) with 12 Intel Xeon L5640 @ 2.27GHz

Algorithm	(1)	(2)	(3)	(4)
Standard implementation with union-find strategy	560	459	158 004	—
Standard implementation of [3]	371	546	7 801	154 595
[3] using union-find strategy	398	556	8 173	77 256
Multithreaded extension of [3]	301	305	4 555	130 286
Proposed singlethreaded version	351	369	11 297	1 098 163
Proposed multithreaded version	**229**	**183**	**3 327**	**76 866**

5 Conclusion

Among hierarchical image representations proposed in the scope of mathematical morphology, the recent α-tree model [6] has already shown great interest in image analysis, e.g. in remote sensing or in video processing. In order for this model to be used in real-time or near real-time context, some efficient computing schemes are required.

In this paper, we address this problem and propose several strategies to achieve an efficient computation of the α-tree model. We compare between single-threaded and multi-threaded architectures and show how the α-tree can be built in a reasonable time. Some preliminary experimental results are provided to illustrate the benefits from our algorithms over existing implementations [3].

The main difference of the method presented here from the other connected component algorithms is a more complex tree structure that allows to construct the tree with minimal depth. For certain distance functions, this significantly lowers the traversal time during the construction. The cost is larger memory consumption of the tree.

Future work will deal with the introduction of more global constraints over the tree, namely the ω dissimilarity or other elements of constrained connectivity [11]. Moreover, we would like to build upon these first results in order to offer a wide range of efficient algorithms for computing tree representations, from the max (or min) tree to more recent hyperconnected trees [7]. We believe this is necessary to disseminate these models within the image processing and computer vision communities.

References

1. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada, vol. 2, pp. 416–425 (July 2001)
2. Matas, P., Dokladalova, E., Akil, M., Georgiev, V., Poupa, M.: Parallel hardware implementation of connected component tree computation. In: 2010 International Conference on Field Programmable Logic and Applications (FPL), August 31-September 2, pp. 64–69 (2010)
3. Merciol, F., Lefèvre, S.: Fast image and video segmentation based on α-tree multiscale representation. In: International Conference on Signal Image Technology Internet Systems, Naples, Italy (November 2012)
4. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Transactions on Image Processing 15(11), 3531–3539 (2006)
5. Ouzounis, G., Syrris, V., Gueguen, L., Soille, P.: The switchboard platform for interactive image information mining. In: Soille, P., Iapaolo, M., Marchetti, P., Datcu, M. (eds.) Proc. of 8th Conference on Image Information Mining, pp. 26–30. ESA-EUSC-JRC (October 2012)
6. Ouzounis, G.K., 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)
7. Perret, B., Lefèvre, S., Collet, C., Slezak, E.: Hyperconnections and hierarchical representations for grayscale and multiband image processing. IEEE Transactions on Image Processing 21(1), 14–27 (2012)
8. 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)
9. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Transactions on Image Processing 7(4), 555–570 (1998)
10. Serra, J.: Anamorphoses and function lattices. In: Dougherty, E.R. (ed.) Mathematical Morphology in Image Processing, ch. 13, pp. 483–523. Marcel Dekker, New York (1993)
11. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(7), 1132–1145 (2008)
12. 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)
13. Soille, P., Najman, L.: On morphological hierarchical representations for image processing and spatial data clustering. In: Köthe, U., Montanvert, A., Soille, P. (eds.) WADGMM 2010. LNCS, vol. 7346, pp. 43–67. Springer, Heidelberg (2012)
14. Tarjan, R.E.: Efficiency of a good but not linear set union algorithm. Journal of the ACM 22, 215–225 (1975)

Ground Truth Energies for Hierarchies of Segmentations

Bangalore Ravi Kiran and Jean Serra

Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, A3SI, ESIEE
{j.serra,kiranr}@esiee.fr

Abstract. In evaluating a hierarchy of segmentations H of an image by ground truth G, which can be partitions of the space or sets, we look for the optimal partition in H that "fits" G best. Two energies on partial partitions express the proximity from H to G, and G to H. They derive from a local version of the Hausdorff distance. Then the problem amounts to finding the cut of the hierarchy which minimizes the said energy. This cuts provide global similarity measures of precision and recall. This allows to contrast two input hierarchies with respect to the G, and also to describe how to compose energies from different ground truths. Results are demonstrated over the Berkeley database.

1 Introduction

Classically, the evaluation of a segmentation w.r.t. a ground truth is viewed as a problem of comparing two partitions of the space E. There are various metrics proposed which are described well surveyed in [4]. The thesis [1] provides refinement tolerant based errors, Local and Global consistency errors(LCE, GCE), due to differences in rendering the ground truth by different human experts. [4] proposes a local region based measure, the segmentation covering, which is the ratio of intersection of 2 classes over the union of their supports, weighted by the relative size of the class w.r.t the input image.

This method is also used to evaluate classes, regions and full partitions of the hierarchy which correspond to the threshold of the Ultrametric contour Map(UCM). As pointed out in [5] the merging order is not the only "cut" in a hierarchy of partitions. The total number of cuts possible consists of the set of partitions formed by the power set of the classes of leaves in the finest partition in a hierarchy.

On the subject of evaluating hierarchies of segmentations there is the work of J. Pont-Tuset and F. Marques [5] closest to this subject of the paper. They determine the upper bound on the correspondences between a ground truth partition and all partitions in a input hierarchy. The comparison performs a global match correspondence between all contours in the hierarchy with respect to the Ground truth partition contours. This thus involves a combinatorial optimization problem, since one must choose a set of contours at various levels and having minimal distance from the ground truth. Indeed, the upper bound

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 123–134, 2013.

introduced in [5] is nothing but the optimal cut in the sense of [8], i.e. the cut which minimizes a given energy, and whose computation is extremely easy as soon as the energy is h-increasing [6]. [9] also propose a local optimization which depends on the number of classes in the cut in a hierarchy with respect to the ground truth segmentation.

The last remark orients us towards the convenient classes of energies acting on hierarchies, as studied in [8]. These energies will be addressed to evaluate hierarchies, a question which covers three aspects:

1. Given a hierarchy H of segmentations of an image I, and a ground truth partition G, how to find a local and a global measures of proximity of the quality of H relatively to G and vice versa, G relative to H ? By local here, we mean a space map of the quality. In fact, we will see that this involves two reciprocal notions. Note that G may be, or not, a contour, but models a drawing by lines and points ($G = \partial G$).
2. When several humans provide several ground truths, how to compose information from multiple ground truth sets G_i? What to do in particular when each drawing concerns a limited zone of the space, which varies with the human/expert?
3. Finally how to evaluate globally the proximities any two given different hierarchies H_1, H_2, with respect to a given common ground truth G.

To summarise symbolically: 1. $H \rightarrow G$ and $G \rightarrow H$, 2. $H \rightarrow G_i$ and vice versa, where G_i refers to a set of ground truths indexed by i, 3. $H_j \rightarrow G$, where H_i refers to different input hierarchies, indexed by j.

After a brief recall of *optimal cuts* and the optimization framework, the above three problems will be successively tackled: the first two by *optimal cuts*, and the third one by means of global similarity measure defined on the saliency function representing the hierarchy. For the sake of pedagogy, we demonstrate on one image, namely the n° 25098 of Berkeley data base, and on the two ground truths depicted in Fig.1, though results are available over available over all images in the database shortly.

2 Reminders

2.1 Hierarchy and Saliency

We start from the definitions used in [8] where,a hierarchy H is a *finite* chain of partitions π_i, i.e.

$$H = \{\pi_i, 0 \leq i \leq n \mid i \leq k \leq n \Rightarrow \pi_i \leq \pi_k\}, \tag{1}$$

The lowest level π_1 is called "the leaves", and the highest one is E itself. An energy ω is associated with each partial partition of E. If $\mathcal{D}(E)$ designates the set of the partial partitions E, then $\omega : \mathcal{D}(E) \rightarrow \mathbb{R}^+$. Let π_1 and π_2 be two partial partitions of same support, and π be a partial partition disjoint from π_1 and π_2.

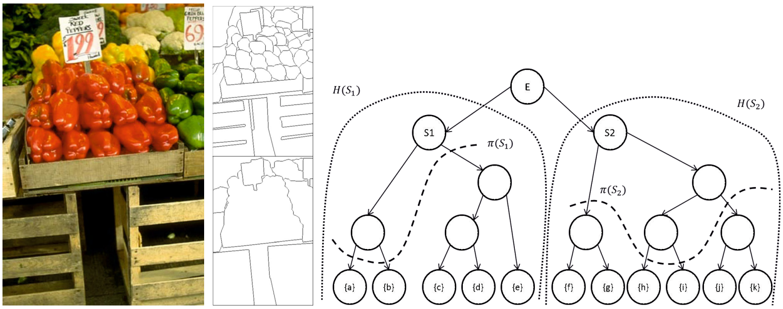


Fig. 1. left: 25098 image from BSD database and 2 of its ground truths GT2 and GT7, right: A hierarchy H with undulating cuts shown $\pi(S_1)$ and $\pi(S_2)$

An energy ω on $\mathcal{D}(E)$ is said to be *hierarchically increasing, or h-increasing* when we have

$$\omega(\pi_1) \leq \omega(\pi_2) \quad \Rightarrow \quad \omega(\pi_1 \sqcup \pi) \leq \ \omega(\pi_2 \sqcup \pi). \tag{2}$$

This condition is necessary and sufficient for obtaining the cut(s) which minimize ω, by running only once through the classes of H in an ascending order. This provides for a dynamic program that only performs a local comparison between a parent class and a composition of its children classes in the hierarchy. The most popular representation of a hierarchy is the dendrogram, shown in (figure). The advantage of this representation is it makes explicit the parent-child relation. Another useful representation, more compact, is the saliency map. It consists in a weighted version of all the edges separating the leaves. Each threshold of the saliency map results in an horizontal cut in the hierarchy. Intuitively, the saliency map is a function that helps visualize the different prominent partitions in the hierarchy.

2.2 Hausdorff Distance

Most of the supervised evaluations of hierarchies, including the present one, and also [2], [4], and [5], derive from the intuition of the Hausdorff distance, in various critical manners. Let us briefly recall this background.

In a metric space E of distance d we aim to match the support $S(\pi)$ of a bounded partial partition π with a set G of points and lines, considered as a ground truth drawing. The smallest isotropic dilation of G that covers the contour $S(\pi)$ has a radius

$$\rho_G = \inf\{\rho \mid G \oplus \rho B \supseteq S(\pi)\}, \tag{3}$$

where ρB is the disc of radius ρ centred at the origin. One can interpret ρ_G as the "energy" required for reaching ∂S from the ground truth G. In the same way, the dual covering is given by the radius ρ_A

$$\rho_A = \inf\{\rho \mid S(\pi) \oplus \rho B \supseteq G\}. \tag{4}$$

By introducing the so called *distance function* $d(x, Z)$ from point x to the fixed set Z, i.e.

$$d(x, Z) = \inf\{d(x, z), z \in Z\} \quad x \in E \tag{5}$$

we see that

$$\rho_G = \sup\{d(x, G), x \in S(\pi)\} \quad \text{and} \quad \rho_A = \sup\{d(x, S(\pi)), x \in G\}, \tag{6}$$

an interpretation which connects the distance function with the partial order on sets by inclusion. In Rel.(6) the value ρ_G (resp. ρ_A) is the maximal distance from a point of ∂S to G (resp. of G to ∂S). The first one, ρ_G, indicates how precise is S w.r.t. the ground truth, the second one, ρ_A, how representative is this ground truth. In indexation, these two numbers are respectively named *precision* and *recall.* The symmetric expression $\rho = \max\{\rho_G, \rho_A\}$ is the well known Hausdorff distance

Hausdorff distance is lacking of finesse because it is a global notion, and of robustness because it uses suprema. If we could define a local equivalent, associated with each class T of π, and no longer with the whole $S(\pi)$ itself, then the regions with a good fit would be treated separately from the others. And in addition, if this equivalent was h-increasing, then it would provide an energy for calculating easily the associated optimal cut [8], i.e. the smallest upper bound of all cuts of the hierarchy, in the wording of [5]).

3 Ground Truth Energy by Local Hausdorff Dissimilarity

In what follows, "best cut", or "optimal cut" must be understood in the sense of "best fitting cut", i.e. the cut which minimizes a given energy of proximity with the ground truth G. It is usually not a criterion of best visual quality.

Precision energy. We now focus on the classes $\{T_i\}$ whose concatenation $T_i \sqcup T_2 ... \sqcup T_k$ generates π. The $\{T_i\}$ are said to be the sons of father S. Consider the class T_i of the partition π. The smallest dilate $G \oplus \rho B$ that covers T_i has a radius

$$\omega_G(T_i) = \inf\{\rho \mid G \oplus \rho B \supseteq T_i\}. \tag{7}$$

By taking the supremum of all $\omega_G(S)$ we find the above value ρ_G of Rel.(3):

$$\rho_G = \bigvee\{\omega_G(S), S \sqsubseteq \pi_A\}. \tag{8}$$

This shows the soundness of ω_G. But a problem arises when we want to extend it from sets to the partial partitions $\mathcal{D}(E)$ of E by some law of composition between theT_i. When the chosen energy is h-increasing, which will always be the case here, finding optimal cuts in hierarchies amounts to compare the partition energies of fathers and sons [8]. If we compose the energies of the sons by supremum, then we trivially always find $\omega_G(\pi) = \omega_G(S)$. If we compose by infimum, we have $\omega_G(\pi) = \omega_G(S)$ when the $\omega_G(T_i)$ all identical, and $\omega_G(\pi) < \omega_G(S)$ when not. And if we compose the energies of the sons by averaging, we obtain again

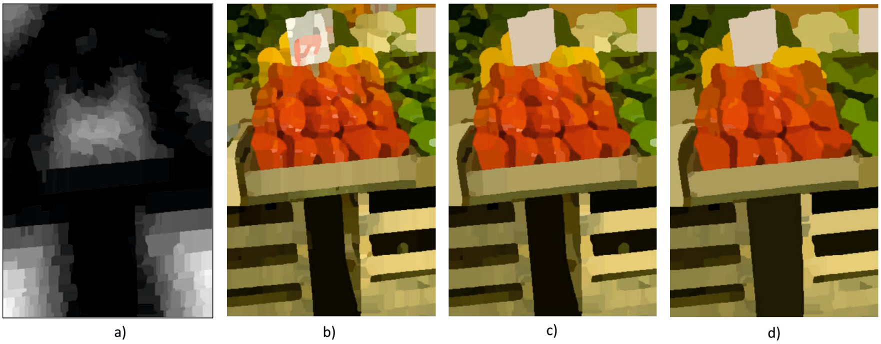


Fig. 2. a) Distribution of the energy ω_G of the leaves classes; b) c) and d) optimal cuts for $\lambda = 0; 10;$ and 20

$\omega_G(\pi) < \omega_G(S)$. Therefore, in all cases, we arrive to an optimal cut which can only be at the lowest level of hierarchy H, i.e. the leaves, or at the highest one, i.e. the space E itself.

For being more informative, we can introduce a trade off based on mutual comparisons of the energies of the sons. An easy way consists in adding a quantizer λ in the composition by infimum, so that

$$\omega_G(\pi) = \omega_G(T_i \sqcup T_2 ... \sqcup T_k) = \inf\{\omega_G(T_i)\} + \lambda. \tag{9}$$

As this new energy is h-increasing, the optimal cut is reached in one pass by comparing the respective energies of sons and fathers [8]. As $\omega_G(S) = \sup\{\omega_G(T_i)\}$, we have $\omega_G(\pi) < \omega_G(S)$ iff $\lambda < \sup\{\omega_G(T_i)\} - \inf\{\omega_G(T_i)\}$. The father replaces the sons when the latter are sufficiently "identical" , i.e. with energy variation $\leq \lambda$. For each value of λ we thus obtain the cut which minimizes the distances to the ground truth G, i.e. the smallest upper bound of all cuts, in the sense [5]. For $\lambda = 0$ we find the leaves partition, and as λ increases, the similar sons w.r.t. their distance to G are progressively clustered, as shown in Fig. **??** of the leaves classes for the ground truth $GT2$; b) c) and d) optimal cuts for $\lambda = 0; 10;$ and 20.

Recall energy. The number $\omega_G(S)$ informs us about those points of ∂S close enough to G, but not on those of G close to ∂S. We cannot take, here, the dual form of the $\omega_G(S)$ of Rel.(7), as we did before with the global Hausdorff distance. Such a dual energy would be

$$\omega'_G(S) = \inf\{\rho \mid S \oplus \rho B \supseteq G\}, \tag{10}$$

a quantity which risks to be extremely large, for the drawing G may spread over the whole space, whereas class S is locally implanted. Fortunately, when dealing with h-increasing energies, one is less interested in the actual values of the energies than by their *increments* between fathers and sons. Now, when a point of G is outside class S, then its distance to S is the same as the max of the distances to the sons T_i of S:

$$x \in G \cap S^c \quad \Rightarrow \quad d(x,S) = d(x,\partial S) = \bigvee d(x,T_i) = \bigvee d(x,\partial T_i), \tag{11}$$

so that the part of G exterior to S is not significant. For the sake of comparison, it thus suffices to focus only on the distances involved in the covering of $G \cap S$ by dilations of ∂S on the one hand, and on those of ∂T_i on the other hand. Then the energy ω'_G of Rel.(10) has to be replaced by the more appropriate one

$$\theta_G(S) = \inf\{\rho \mid S \oplus \rho B \supseteq G \cap S\}. \tag{12}$$

When S spans all classes of a partition π_A, then the supremum of all $\theta_G(S)$ gives the value ρ_A of Rel.(4)

$$\rho_A = \bigvee\{\theta_G(S), S \sqsubseteq \pi\}, \tag{13}$$

and the (global) Hausdorff distance ρ between π and G turns out to be the double supremum,

$$\rho = \bigvee\{\{\omega_G(S)\bigvee\{\theta_G(S)\}, S \sqsubseteq \pi\}. \tag{14}$$

It remains to verify that θ_G is h-increasing.

Proposition 1. *Given a ground truth set G, the extension of the energy θ_G of Rel.(12) to partial partitions by $\vee$ composition is h-increasing.*

Proof. Let $\pi(S_1)$ and $\pi'(S_1)$ be two partial partitions of set S_1, with

$$\theta_G(\pi(S_1)) = \bigvee\{\theta_G(T_i), T_i \sqsubseteq S\} \leq \theta_G(\pi'(S_1)) = \bigvee\{\theta_G(T'_i), T'_i \sqsubseteq S'_1\} \tag{15}$$

Consider a partial partition $\pi(S_2)$, where $S_2 \subseteq S_1^c$. By taking the supremum of each member of inequality (15) with $\bigvee\{\theta_G(X_j), X_j \sqsubseteq S_2\}$ one does not change the sense of the inequality, which becomes

$$\theta_G(\pi(S_1) \sqcup \pi(S_2)) \leq \theta_G(\pi'(S_1) \sqcup \pi(S_2)), \tag{16}$$

which achieves the proof.

Note that when $G \cap S = \varnothing$, then $\theta_G(S) = 0$.

Composition of $\omega_G(S)$ and $\theta_G(S)$. The composition of the energies happens with respect to a single ground truth, or to several ones. In the first case, one can wonder if preferable not to combine ω_G and θ_G so that they can provide two separated maps for the precision and for the recall. The two associated overall values may be presented in a 2-D graphic as proposed in [3]. We can also take for final energy either max(ω_G, θ_G), or sum ω_G+ θ_G, they are both h-increasing. On the example of the "peppers", and for two different ground truths, one obtains the results depicted in Fig.3

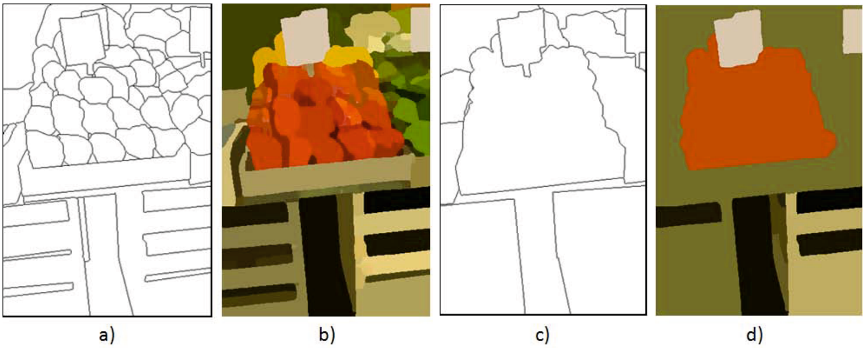


Fig. 3. a) and c), ground truth $GT2$ and $GT7$; b) and d) energies ω_G+ θ_G, for $G = GT2$ and $G = GT7$

In case of multiple ground truths, the usual techniques proposed in literature are additive. Formally speaking, why not? Putting $\omega_G = \sum \omega_{G_i}$ yields an h-increasing energy, hence a best cut (which is, of course different from the sum of the best cuts of the various G_i). The implicit assumption here is that all ground truths are more or less similar.

But one can also encounter drawings G_i that focus on different regions of the scene. Then if we take the sum, each part of the space risks to be penalized because if is far from one drawing, at least.

For the situation depicted in Fig.4, the energies first two best cuts are given by $\sup\{\omega_G, \theta_G\}$ and the third one by taking $\inf\{\sup\{\{\omega_{G_1}, \theta_{G_1}\}, \sup\{\{\omega_{G_2}, \theta_{G_2}\}\}$. When point $x \in E$ is farther from G_1 than from G_2 then the G_1energy is not taken into account.

4 Other Energies

Conditional energy. The two energies $\omega_G(S)$ and $\theta_G(S)$ of Rel. (12) and (9) have been chosen because of their geometrical meanings, but they are far for being the only possible ones. It is iindeed easy to build an energy which fits the features one wants to emphazise. Suppose for example that we decide that the number of classes n of the ground truth is a cruxial feature. Then when applying energy ω_G we can condition the ascending pass which generates the best cut to stop as soon as the number n of classes is reached. Fig. 5 depicts the best cuts w.r.t. ω_G. when the parameter λ of Eq.(9) equals $0, 10$, and 80, and when the ground truth is GT7, which has 87 classes. For $\lambda = 0$, we do not obtain the leaves partition, because the classes with an equal energy have been clustered, as pointed out previously. In Fig. 5c) and d), but not in Fig. 5b), one arrives to 87 classes before the end of the climbing algorithm. This explains why the two partitions are not comparable.

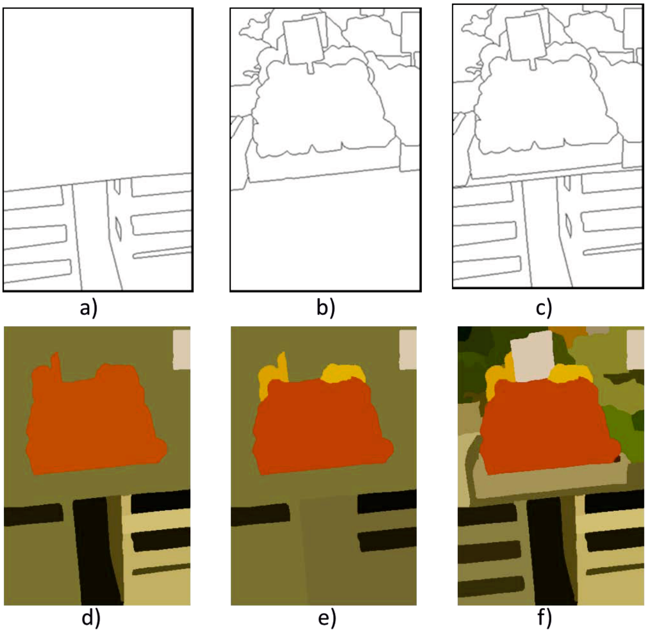


Fig. 4. a),b), and c) two ground truths and their union; d),e), and f) the coresponding optimal cuts

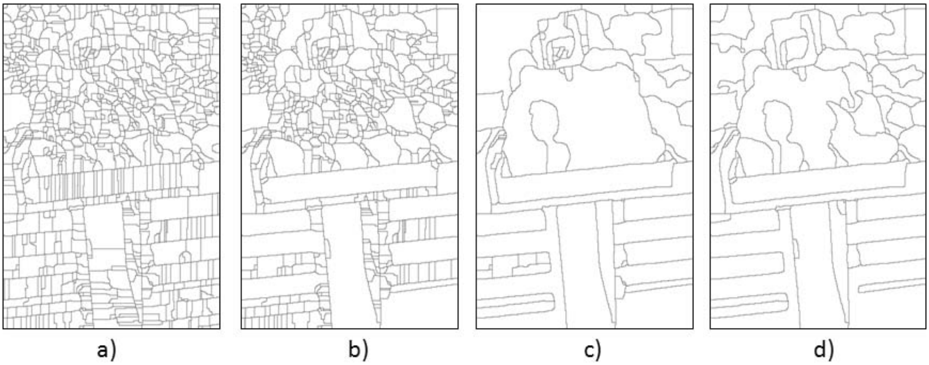


Fig. 5. a) Leaves of hierarchy b), c) and d) Conditional best cuts for $\lambda = 0$, 10 and 80

Local linear dissimilarity. Another variant consists in replacing the supremum that appears in Rel.(6) by a L^p sum, which gives less importance to the farthest zones. A similar approach has been successfully used by L. Gorelick et Al. [12] in regional line-search cuts. Among the L^p integrals, the one which weakens the most the weights of the farthest zones is obtained for $p = 1$. Therefore we take for precision energy $\widetilde{\omega}_G(S)$ the integral of distance function $g(x)$ of G along the contour ∂S and for recall energy $\widetilde{\theta}_G(S)$ the integral of the distance function $g(x, \partial S)$ of S on $G \cap S$:

$$\widetilde{\omega}_G(S) = \frac{1}{\partial S} \int_{\partial S} g(x)dx \qquad \widetilde{\theta}_G(S) = \frac{1}{G \cap S} \int_{G \cap S} g(x, \partial S)dx \tag{17}$$

The two functionals $\widetilde{\omega}_G$ and $\widetilde{\theta}_G$ are extended from classes to partial partitions by addition, since they both involve integrals, and one easily checks that the two energies are h-increasing. The higher $\widetilde{\omega}_G(S)$, (resp.$\widetilde{\theta}_G(S)$), the farther S is from G (resp. G is from S). In case of a ground truth given by k drawings, one just sums up the k energies $\widetilde{\omega}_G$ and $\widetilde{\theta}_G$.

5 Global Measures of Precision and Recall for Hierarchies

Following from the local measures in (17) which are integrals of the distance function associated with each class, we propose here a global similarity measures for a hierarchy. Two global measures of precision and recall for a given hierarchy H of segmentations with respect to an input ground truth partition G. The measure now is not between 2 partitions any more and deals with the global similarity between hierarchies of partitions H and a single partition G. The representative functions we are going to use for the global measures are: s the saliency and g the distance function, the set S_i saliency map threshold at an index i.

$$P = \sum_{i=0}^{1} \frac{i}{N} \frac{\int_{x \in \epsilon(S_i)} (1 - g(x)).S_i(x)\mathrm{d}x}{|S_i|} \quad R = \sum_{i=0}^{1} \frac{i}{N} \frac{\int_{x \in G} (1 - g_{S_i}(x))\mathrm{d}x}{|G|} \tag{18}$$

The integral calculates the similarity between partition S_i produced by thresholding the saliency s at i and the ground truth partition G by integrating the inverse distance function $1 - g$ under the binary function S_i. Also the sense of the hierarchy is such that $s_{i+1} \subset s_i$ which represents that partition at a higher level in the hierarchy has fewer contours than the one below to respect the inclusion order. Each integral is weighted by the relative rank of the partition within the hierarchy H. This is done by weight it by ratio of threshold index i and the total number of levels in the hierarchy N as shown in equation(18).

Similarly a global precision value for the contours of the partitions in the hierarchy can be calculated by integrating the distance functions g_{S_i} of partition S_i under the ground truth partition G. These integrals are normalized with respect to each image support by dividing by the size of the image.

5.1 Proximity between Hierarchies

The integrals in equation (18) is between a partition G (ground truth) and a hierarchy H. The same can be extended to measure the proximities between two hierarchies of partitions. Given two hierarchies of partitions, H_1, H_2, with N and M number of levels, and partitions indexed by i and j respectively,

$$\phi_{12} = \sum_{j \in [1,M]} \sum_{i \in [1,N]} \frac{\int_{x \in \epsilon(\pi_i)} (1 - g_{\pi_j}(x)).\pi_i(x)\mathrm{d}x}{|\pi_i|} \tag{19a}$$

$$\phi_{21} = \sum_{i \in [1,N]} \sum_{j \in [1,M]} \frac{\int_{x \in \epsilon(\pi_j)} (1 - g_{\pi_i}(x)).\pi_j(x)\mathrm{d}x}{|\pi_j|} \tag{19b}$$

where g_{π_i} is the distance function of the partition π_i.

The measure lacks the similarity measures across partitions which are not horizontal cuts, but generally cuts from the two hierarchies. This becomes again a combinatorial problem. The refinement of cuts π_i from an input hierarchy H_1 would have a value of the distance function g_{π_j} which decreases on average till the point where the two partitions nearly fit and the integral starts increasing again.

6 Results

To demonstrate the inputs of the optimization, we show the 2 ground truths used (GT2 and GT4 from the list for image 239096), we show the distribution of ground truth energies(radii) at different thresholds of the saliency. The gray level corresponds to the radius of dilations ω_G and θ_G in Figure 7. for image 239096. Their optimal cuts based on the haussdorf energy corresponding to the supremum of the two radii ω_G and θ_G are shown in Figure 8. We observe that the optimization introduces small parasite classes which are chosen since the children or always more optimal than the parent in certain symmetries.

We evaluate the global measure on 3 hierarchies Arbelaez, Cousty and random hierarchies (generated by merging classes of the leaves partition randomly) for the 2 hierarchies produced from random permutation matrices used as distance functions as explained in the previous section. We evaluate all 7 ground truths w.r.t to the 3 input hierarchies, producing a table (9) of global measures. The P and R measures are averaged over the 7 ground truths available for 25098 image.

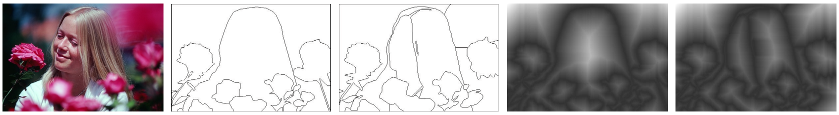


Fig. 6. Input Image 2390986, Ground truths GT2 and GT4, and their distance functions

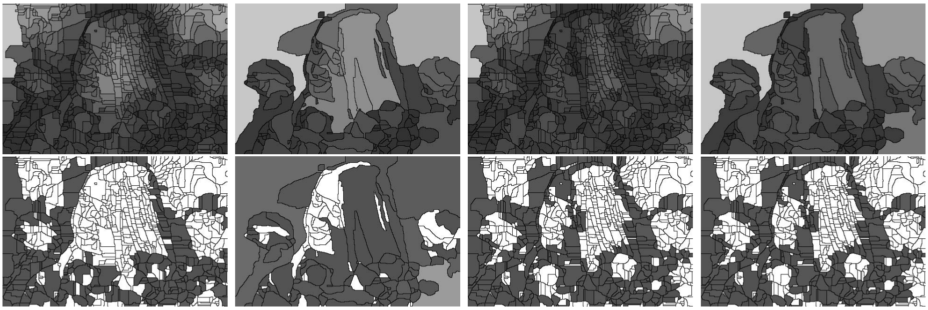


Fig. 7. The distribution of ω_G (top) for threshold of the (UCM) at 0 (leaves), 0.1, for ground truth GT 2(two images on the left) and GT 4(right), the image is contrasted to see the low level values clearer. Same for θ_G on the bottom line.

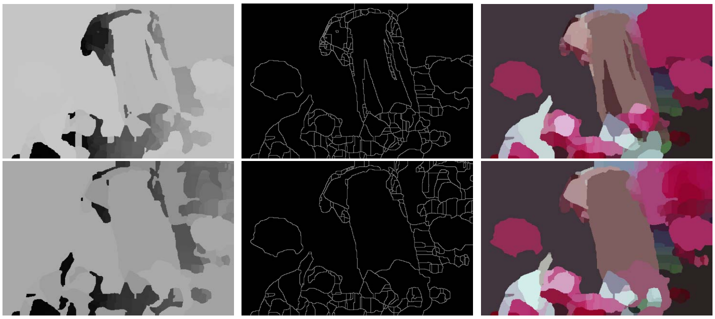


Fig. 8. The Energy distribution for the optimal cut by $\vee(\omega_G, \theta_G,)$ The partition and the image

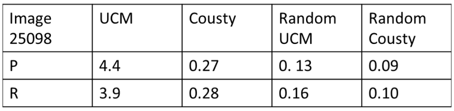


Image 25098	UCM	Cousty	Random UCM	Random Cousty
P	4.4	0.27	0. 13	0.09
R	3.9	0.28	0.16	0.10

Fig. 9. Integrals from equation(18) Expressed per 1000 pixels in the image

7 Conclusion

A method for comparing a hierarchy of partitions H with one, or more, ground truth set G was proposed. Two points of view were developed. The first one is based on the idea of associating two energies that express the proximity between G and H. It was shown that several different criteria, and several laws of composition of the partition's classes lead to emphasize different aspects of the hierarchy. The same approach permits also to combine different ground truths associated with different zones of the space. Finally a global similarity measure is used to evaluate the proximity of hierarchies of partitions and a ground truth partition. Future work would consist in using other image feature based energies and studying the law of compositions.

References

1. Martin, D.R.: An Empirical Approach to Grouping and Segmentation, PhD Thesis, EECS Department, University of California, Berkeley, Number = UCB/CSD-03-1268 (2003)
2. Arbeláez, P.: Une approche mtrique pour la segmentation d'images, Phd thesis, Univ.of Paris Dauphine (November 2005)
3. Arbeláez, P., Cohen, L.: Constrained Image Segmentation from Hierarchical Boundaries. In: CVPR (2008)
4. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour Detection and Hierarchical Image Segmentation. IEEE PAMI 33 (2011)
5. Pont-Tuset, J., Marques, F.: Supervised Assessment of Segmentation Hierarchies. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 814–827. Springer, Heidelberg (2012)
6. Serra, J.: Hierarchies and Optima. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) DGCI 2011. LNCS, vol. 6607, pp. 35–46. Springer, Heidelberg (2011)
7. Serra, J., Kiran, B.R.: Climbing the pyramids CoRR abs/1204.5383 (2012)
8. Serra, J., Kiran, B.R., Cousty, J.: Hierarchies and climbing energies. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) CIARP 2012. LNCS, vol. 7441, pp. 821–828. Springer, Heidelberg (2012)
9. Pont-Tuset, J., Marqués, F.: Upper-bound assessment of the spatial accuracy of hierarchical region-based image representations. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (2012)
10. Movahedi, V., Elder, J.H.: Design and perceptual validation of performance measures for salient object segmentation. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 49–56 (2010)
11. Cousty, J., Najman, L.: Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 272–283. Springer, Heidelberg (2011)
12. Gorelick, L., Schmidt, F.R., Boykov, Y., Delong, A., Ward, A.: Segmentation with non-linear regional constraints via line-search cuts. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 583–597. Springer, Heidelberg (2012)

Playing with Kruskal: Algorithms for Morphological Trees in Edge-Weighted Graphs

Laurent Najman, Jean Cousty, and Benjamin Perret

Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, A3SI, ESIEE

Abstract. The goal of this paper is to provide linear or quasi-linear algorithms for producing some of the various trees used in mathemetical morphology, in particular the trees corresponding to hierarchies of watershed cuts and hierarchies of constrained connectivity. A specific binary tree, corresponding to an ordered version of the edges of the minimum spanning tree, is the key structure in this study, and is computed thanks to variations around Kruskal algorithm for minimum spanning tree.

1 Introduction

In the theoretical companion paper [1] of the present paper, we show how some morphological hierarchies [2–7] defined on an edge-weighted graph $G = (V, E, F)$ are related, and when it is possible, how they can be computed one from each other. In this paper, we provide efficient quasi-linear or linear algorithms to compute those hierarchies. In particular, this paper contains:

- provided that the edges are either already sorted or can be sorted in linear time, a quasi-linear $O(|E| \times \alpha(|V|))$ (where $\alpha()$ is the extremely slowly growing inverse of the single-valued Ackermann function) algorithm that computes a *binary partition tree by altitude ordering*, a fundamental structure that we post-process in the sequel;
- a linear $O(|V|)$ post-processing algorithm that computes the hierarchy of quasi-flat zones [1, 3] (also know as the α-tree [8]);
- a linear $O(|V|)$ post-processing algorithm that computes (hierarchies of) watershed cuts [9];
- a linear $O(|V|)$ post-processing algorithm that computes hierarchies by increasing attributes; as detailed here and in [1], such an algorithm can be used to obtain either constrained connectivity hierarchies [10] or watershed-based hierarchies [11].

To the best of our knowledge, the only published constrained connectivity algorithm, available in [10], has an unknown complexity and only computes one level of the hierarchy. An algorithm computing the whole hierarchy, relying on the component tree of the edge-weighted graph G, is roughly sketched in [12], but has a complexity higher than the one proposed in this paper and is less memory efficient. Concerning attribute-based hierarchies, the most efficient algorithm [13] has a complexity higher than the one proposed in this paper, and is less efficient.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 135–146, 2013.

At the heart of our approach is the minimum spanning tree (MST): this tree T is a connected spanning graph of the graph G such that the weight of T: $F(T) := \sum_{e \in E(T)} F(e)$ is the least possible weight for a spanning graph of G. As detailed in section 2.1, we rely on Kruskal algorithm [14] for computing this MST. However, while producing the MST, we make use of another tree (detailed in section 2.2) that we call the *binary partition tree by altitude ordering.* Using Tarjan union-find (section 2.3), we propose in section 2.4 an efficient algorithm to compute this binary tree. Post-processing will be studied in section 3.

2 Binary Partition Tree and Minimum Spanning Tree

2.1 Kruskal Algorithm

Kruskal's algorithm [14] is a greedy algorithm that finds a minimum spanning tree for a connected weighted graph. It can be described as follows:

- create a forest $\mathcal{F}$ (a set of trees), where each vertex in the graph is a separate tree
- create a set S containing all the edges in the graph
- while S is nonempty and $\mathcal{F}$ is not yet a single tree
  - remove an edge with minimum weight from S
  - if that edge connects two different trees, then add it to the forest, combining two trees into a single tree
  - otherwise discard that edge.

At the termination of the algorithm, the forest has only one component and forms a minimum spanning tree of the graph.

The efficiency of Kruskal's algorithm relies on a disjoint-set data structure that keeps track of a set of elements partitioned into a number of disjoint (non-overlapping) subsets. The disjoint set problem consists in maintaining a collection $\mathcal{Q}$ of disjoint sets under the operation of union. Each set Q in $\mathcal{Q}$ is represented by a unique element of Q, called the *canonical element.* In the following, q_1 and q_2 denote two distinct elements. Three operations allow to manage the collection:

- `MakeSet(`q_1`)`: add a new element q_1 to the collection $\mathcal{Q}$, provided that the element q_1 does not already belongs to a set in $\mathcal{Q}$.
- `FindCanonical`(q_1): return the canonical element of the set in $\mathcal{Q}$ which contains q_1.
- `Union`(q_1, q_2): let Q_1 and Q_2 be the two sets in $\mathcal{Q}$ whose canonical elements are q_1 and q_2 respectively (q_1 and q_2 must be different). Both sets are removed from $\mathcal{Q}$, their union $Q_3 = Q_1 \cup Q_2$ is added to $\mathcal{Q}$ and a canonical element for Q_3 is selected and returned.

An implementation of Kruskal algorithm is presented in Algorithm 1. In this implementation, we identify any element of V with an integer corresponding to its index in the finite set $|V|$. We save the edge of the MST in an array `MST`, hence obtaining a strict order on the edges; this order is necessary for post-processing.

Algorithm 1. Kruskal

```
Data: An edge-weighted graph $(V, E, F)$.
Result: A minimum spanning tree MST
Result: A collection $\mathcal{Q}$
// Collection $\mathcal{Q}$ is initialized to $\emptyset$
$e := 0$;
for all $x_i \in V$ do MakeSet($i$);
for all edges $\{x, y\}$ by (strict) increasing weight $F(\{x, y\})$ do
    $c_x := \mathcal{Q}$.FindCanonical($x$); $c_y := \mathcal{Q}$.FindCanonical($y$);
    if $c_x \neq c_y$ then
        $\mathcal{Q}$.Union($c_x$, $c_y$);
        MST[e] := $\{x, y\}$;
        $e := e + 1$;
    else DoSomething($\{x, y\}$);
```

When computing an MST, the `DoSomething` procedure does nothing, *i.e.*, it discards the considered edge. In section 3.3, we show an example when the procedure `DoSomething` is useful.

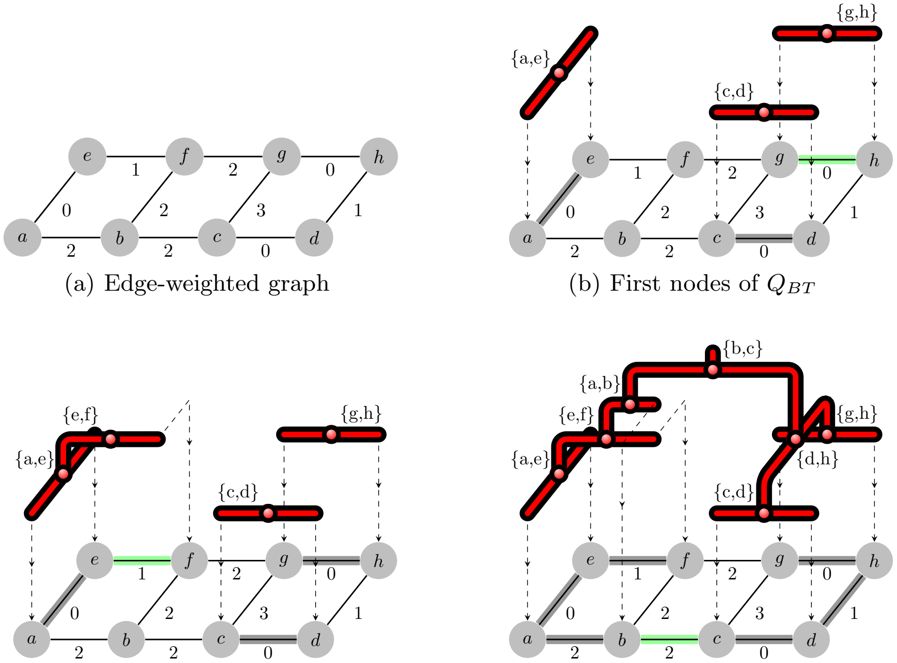
Procedure DoSomethingMST($\{x, y\}$)

```
// Ignore $\{x, y\}$
```

The main question in implementing Kruskal's algorithm is thus how to represent and implement the collection $\mathcal{Q}$. A good representation is to maintain the collection as a set of trees, *i.e* each element of $\mathcal{Q}$ is a tree. In the sequel of this paper, each set of the collection $\mathcal{Q}$ is represented by a rooted tree, where the canonical element of the set is the root of the tree. We are going to play with various tree representations of connected components. Kruskal's algorithm will not change, but different implementations for `Union` and `FindCanonical` will lead to two different trees, one of them being useful for connected filtering.

2.2 A Simple Algorithm That Yields a Binary Partition Tree by Altitude Ordering

The first tree we present is the useful one, although the proposed algorithm in this section is not an efficient one (we make a better proposal in section 2.4). As shown in [1] and in the sequel of this paper, post-processing this tree provides the min-tree of MST, the tree of quasi-flat zones and trees of watersheds. The main idea is the following: each time an edge $\{x, y\}$ is put into the MST, *i.e.* each time a union is made, we create a new node whose children are the two disjoint sets containing x and y. Intuitively, it is as if we break the edge in two, and add a node between the two points of the edge. The added node becomes the canonical element of the union of these two points (see Fig. 1.a, b, c and d).



(a) Edge-weighted graph

(b) First nodes of Q_{BT}

(c) Adding a node to Q_{BT}

(d) Final Q_{BT}

Fig. 1. A simple process for obtaining Q_{BT}, a binary tree providing a strict total order relation on the edges of the MST (see text)

Each node will thus either correspond to an edge of the MST or to a vertex of V. In the implementation, nodes of the tree are represented by integer: nodes corresponding to vertices are from 0 to $|V|-1$ and nodes corresponding to edges of the MST are from $|V|$ to $2|V|-2$. Such a numbering allows the tree itself to be provided thanks to an array `parent` that stores the parent of a given node. If node i does not have a parent, then $parent[i] := -1$. This algorithm is implemented in Q_{BT}.MakeSet, Q_{BT}.FindCanonical and Q_{BT}.Union.

Procedure Q_{BT}.MakeSet(q)

```
$Q_{BT}$.parent[$q$] := $-1$; $Q_{BT}$.size += 1;
```

Function Q_{BT}.FindCanonical(q)

```
while $Q_{BT}$.parent[$q$] $\geq 0$ do $q$ := $Q_{BT}$.parent[$q$];
return $q$;
```

```
Function $Q_{BT}$.Union($c_x, c_y$)
$Q_{BT}$.parent[$c_x$]:=$Q_{BT}$.size; $Q_{BT}$.parent[$c_y$]:=$Q_{BT}$.size;
$Q_{BT}$.MakeSet($Q_{BT}$.size);
return $Q_{BT}$.size-1;
```

At the end, we obtain a collection Q_{BT} which is a binary tree. The Q_{BT} tree provides the order in which the edges have been put into the MST, the latest edge added to the tree being the root of the tree, *i.e.*, the highest one. In other words, the edges of the MST are strictly ordered by Q_{BT}, according to their altitude in the tree. We say that Q_{BT} is a *binary partition tree by altitude ordering* [1]. The complexity of Kruskal implemented with this specific union-find is in $O(|V|^2)$, and thus, this process is not very efficient.

2.3 Efficient MST Implementation with Tarjan Union-Find

Tarjan [15] proposed a very simple and very efficient algorithm called *union-find* to achieve any intermixed sequence of union and find. The implementation of this algorithm is given in procedure Q_T.MakeSet and functions Q_T.Union and Q_T.FindCanonical. To each element of the collection is associated a parent (as precedently) and a rank 'Rnk'. Both the mapping 'parent' and the mapping 'Rnk' are represented as arrays in memory. One of the two key heuristics to reduce the complexity is a technique called *path compression*, that was used by Tarjan to reduce the cost of FindCanonical. It consists, while searching for the root r of the tree which contains q, in considering each element p of the path from q to r (including q), and setting the parent of p to be r. The other key technique, called *union by rank*, consists in always choosing the root with the greatest rank to be the representative of the union while performing the Union operation. The rank Rnk(c_x) of a canonical element c_x is a measure of the depth of the tree rooted in c_x, and is exactly the depth of this tree if the path compression technique is not used jointly with the union by rank technique. If the two canonical elements c_x and c_y have the same rank, then one of the elements, say c_y, is chosen arbitrarily to be the canonical element of the union: c_y becomes the parent of c_x; and the rank of c_y is incremented by one. Union by rank avoids creating degenerate trees, and helps keeping the depth of the trees as small as possible. For a more detailed explanation and complexity analysis, see Tarjan's paper [15].

```
Procedure $Q_T$.MakeSet($q$)
$Q_T$.parent[$Q_T$.size] := $-1$;  $Q_T$.Rnk[$Q_T$.size] := 0; $Q_T$.size += 1;
```

```
Function $Q_T$.FindCanonical($q$)
$r := q$;
while $Q_T$.parent[$r$] $\geq 0$ do $r$ :=$Q_T$.parent[$r$];
while $Q_T$.parent[$q$] $\geq 0$ do $tmp := q; q$ :=$Q_T$.parent[$q$]; $Q_T$.parent[$tmp$] := $r$;
return $r$;
```

```
Function $Q_T$.Union($c_x$, $c_y$)
if ($Q_T.Rnk[c_x] > Q_T.Rnk[c_y]$) then swap($c_x, c_y$);
if ($Q_T.Rnk[c_x] == Q_T.Rnk[c_y]$) then $Q_T$.Rnk[$c_y$] += 1;
$Q_T$.parent[$c_x$] := $c_y$;
return $c_y$;
```

As stated at the beginning of this section, implementing Kruskal's algorithm with Tarjan Union-Find leads to a quasi-linear complexity: provided that the edges are either already sorted or can be sorted in linear time (for example with counting sort or radix sort), the complexity is $O(|E| \times \alpha(|V|))$, where $\alpha()$ is the extremely slowly growing inverse of the single-valued Ackermann function. Unfortunately, the tree built by Tarjan Union-Find is of no use for connected filtering: the path compression technique flattens the tree, and does not preserve the order by which the edges or the vertices are processed in the algorithm. In the next section, we are going to combine the two proposals into an efficient one.

2.4 An Efficient Algorithm That Yields a Binary Partition Tree by Altitude Ordering

The function Q_{BT}`.FindCanonical` is slow because it takes some time to find the canonical element for a connected component: we have to climb the tree until the root is found. We can use Tarjan Union-Find on Q_{BT} itself, *i.e.* we can use Q_T as a second collection that maintains a compressed representation of Q_{BT}, so that finding the canonical element is now in quasi-constant time. We also have to store the root of any tree in Q_{BT} in an array *Root* so that such a root can be found in constant time from any tree in Q_T. The implementation uses Q_{EBT}`.MakeSet`, Q_T`.FindCanonical` and Q_{EBT}`.Union`.

```
Procedure $Q_{EBT}$.MakeSet($q$)
$Q_{EBT}$.Root[$q$]:=$q$; $Q_{BT}$.MakeSet($q$); $Q_T$.MakeSet($q$);
```

```
Function $Q_{EBT}$.Union($c_x, c_y$)
$t_u$:=$Q_{EBT}$.Root[$c_x$]; $t_v$ := $Q_{EBT}$.Root[$c_y$];
// Union in $Q_{BT}$(without compression)
$Q_{BT}$.parent[$t_u$] := $Q_{BT}$.parent[$t_v$] := $Q_{BT}$.size;
// If children are needed, add them to the root
$Q_{BT}$.children[$Q_{BT}$.size].add({$t_u$}); $Q_{BT}$.children[$Q_{BT}$.size].add({$t_v$});
c:=$Q_T$.Union($c_x$,$c_y$); // Union in $Q_T$(with compression)
$Q_{EBT}$.Root[c] := $Q_{BT}$.size; // Update the root of $Q_{EBT}$
$Q_{BT}$.MakeSet($Q_{BT}$.size);
return $Q_{BT}$.size-1;
```

```
Function $Q_{EBT}$.FindCanonical($q$)
return $Q_T$.FindCanonical($q$);
```

When Kruskal is finished, the tree Q_{BT} is exactly the same as the one in section 2.2. The only difference is thus the speed of the algorithm: thanks to the use of Tarjan Union-Find, the complexity of this Kruskal algorithm using Q_{EBT} is quasi-linear $O(|E| \times \alpha(|V|))$.

3 Post-Processing the Binary Tree

In this section, we are going to detail some linear $O(|V|)$ algorithms that produce, from Q_{BT}, (1) a watershed cut, (2) a hierarchy of quasi-flat zones, and (3) any attribute-based hierarchy (if the attribute is increasing).

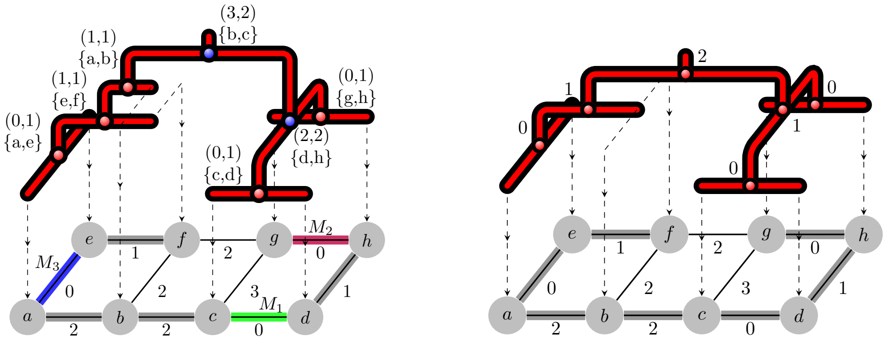
As we have seen, each node of Q_{BT} corresponds either to a vertex of the graph or to an edge of the MST. Recall that edges of MST are sorted by a strict order relationship that follows increasing weight-edges: the $|V|$ first nodes of Q_{BT} are the vertices of $|V|$, and the root of Q_{BT} corresponds to the edge of the MST with the greatest weight. To ease the reading of the algorithms of this section, we provide below two helper functions that clarify how we can pass from the nodes of Q_{BT} to the edges of the MST and how to obtain the weight of the edge of the MST corresponding to a given node of Q_{BT}.

```
Function getEdge($n$)
  Data: a (non-leaf) node $n$ of $Q_{BT}$
  Result: the edge $e$ of the MST corresponding to the $n^{th}$ node
  return $n - |V|$;
```

```
Function weightNode(n)
  Data: a (non-leaf) node of the tree
  Result: the weight of the MST edge associated with the $n^{th}$ node of $Q_{BT}$
  return F(MST[getEdge(n)]);
```

3.1 Watershed Cuts

A watershed cut [9] can be easily obtained in our framework. Indeed, those cuts are the "highest separations" of minima of the minimum spanning tree. We propose below a linear algorithm `watershed` that labels the edges of the MST with a flag stating whether or not an edge is a watershed edge. The main idea is to obtain the latest edge of the MST included in a given minimum in Q_{BT}. This can be done by counting the number of minima of the edge-weighted graph (G, F) thanks to a simple process than run through the nodes of Q_{BT} by increasing order, and checks whether or not a given node has an altitude lower than its parent and does not contain a minimum. We increment a counter for the ancestors of several minima. An edge is a watershed edge if it merges several catchment basins corresponding to different minima. The complexity of the function `watershed` is linear in the number of vertices, and thus the whole process that computes a watershed cut is quasi-linear.



(a) Watershed

(b) Q_{CT}

Fig. 2. Two trees derived from Q_{BT} by post-processing. (a) A hierarchical tree Q_{BT} on which watershed edges have a blue point. Edges M_1 (green), M_2 (red) and M_3 (blue) are the minima of the edge-weighted graph. (b) The tree of the quasi-flat zones hierarchy, a canonized version of Q_{BT} (*i.e.*, the min-tree of the MST.)

```
Function watershed
Data: $Q_{BT}$
Result: A binary array $ws$ indicating which MST edges are watershed
for all leaf-nodes $n$ of $Q_{BT}$ do minima[$n$]:=0;
for each non-leaf node $n$ of $Q_{BT}$ by increasing order do
    flag := TRUE; $nb$ := 0;
    for all $c \in Q_{BT}.children[n]$ do
        $m$ := minima[c]; $nb := nb + m$;
        if ($m == 0$) then flag := FALSE;
    ws[getEdge(n)] := flag;
    if ($nb \neq 0$) then minima[n] := $nb$;
    else
        if (n is the root of $Q_{BT}$) then minima[n] := 1;
        else
            p := $Q_{BT}$.parent[n];
            if (weightNode[n]<weightNode[p]) then minima[n] := 1;
            else minima[n]:=0;
```

The set of watershed edges provides a MST of the neighborhood graph of the catchment basins [16]. In Fig. 2.a, the nodes of Q_{BT} corresponding to watershed edges have a blue point.

By removing from Q_{BT} all nodes that are not watershed ones, we obtain a filtered tree that corresponds to a hierarchy of watershed cuts, more precisely the one corresponding to an ultrametric watershed [12]. In the sequel of this paper, one can use indifferently either the original Q_{BT} or this modified one. In the first case, one is working within the framework of the constrained connectivity, and in the second case, one is working within the framework of watershed cuts.

One of the interests of working with watershed cuts rather than with flat zones is that the hierarchy is smaller, as the super-pixels (catchment basins) provided by watershed cuts are larger than the super-pixels provided by the flat-zones. From a practical point of view, greater speed can thus be achieved thanks to watershed cuts.

3.2 Quasi-flat Zones Hierarchy

In this section, we post-process Q_{BT} to obtain the tree of the quasi-flat zones hierarchy [3], which is proved [1] to be Q_{CT}, the min-tree of the minimum spanning tree. The differences between Q_{BT} and Q_{CT} are illustrated in Fig. 1.d and 2.b. Q_{CT} can be computed directly thanks to a dedicated Union-Find procedure that we will describe in an extended version of this paper. Here, due to space constraints, we only propose a short post-processing that transforms Q_{BT} into the desired min-tree Q_{CT}. The implementation uses CanonizeQ_{BT} to post-process Q_{BT}, and needs the children of a node.

```
Procedure Canonize$Q_{BT}$
  Data: $Q_{BT}$
  Result: $Q_{CT}$, a canonized version of $Q_{BT}$
  for all nodes n of $Q_{BT}$ do $Q_{CT}$.parent[n]:=$Q_{BT}$.parent[n]; $Q_{CT}$.size+=1;
  for each non-leaf and non-root node n of $Q_{BT}$ by decreasing order do
      $p := Q_{CT}$.parent[$n$];
      if (weightNode(p) == weightNode(n)) then
          for all $c \in Q_{BT}$.children[$n$] do $Q_{CT}$.parent[$c$]:=$p$;
          $Q_{CT}$.parent[$n$]:=$n$; // Delete node $n$ of $Q_{CT}$
  // If needed, build the list of children
  for all nodes n of $Q_{CT}$ do
      p:=$Q_{CT}$.parent[n]; if $p \geq 0$ and $p \neq n$ then $Q_{CT}$.children[p].add(n);
```

The procedure CanonizeQ_{BT} is in $O(|V|)$, and thus the whole process that computes a quasi-flat zones hierarchy is quasi-linear. To the best of our knowledge, this is the most efficient algorithm published for computing this hierarchy. However, for most image-processing tasks, the binary partition tree Q_{BT} can be used instead, and this is what we are going to do in the sequel of this paper.

3.3 Attribute-Based Hierarchies

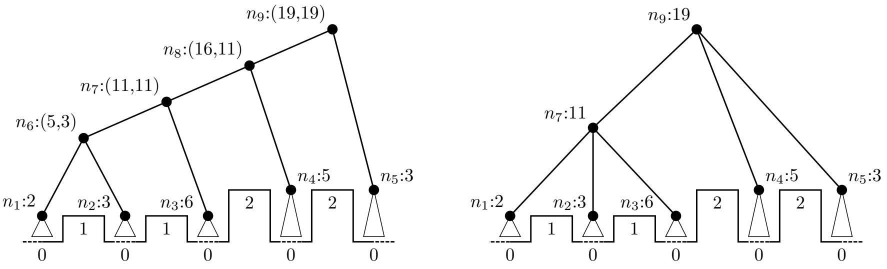
Attributes It is easy to compute some attributes on each node of Q_{BT}: surface (number of vertices in a node), depth, volume or ordered markers are the most classical attributes [11]. Another attribute from the constrained connectivity framework [10] is the range. Any of those previous attributes are *increasing*: recall that an attribute A is increasing if when there is a parenthood relationship between two nodes n_1 and n_2, *i.e.* when the vertices of n_1 are contained in the vertices of n_2, then $A(n_1) \leq A(n_2)$. In the sequel of this section, we denote by `attributeComp`$[n]$ the attribute of the node n of Q_{BT}.

It is when computing an attribute that the procedure `DoSomethingMST` can be useful: for example, one can, using `DoSomethingMST`, register each edge of the graph at the correct place in Q_{BT}, by adding new nodes corresponding to these edges. Then, for example, we can imagine using an attribute such as the surface computed not from the vertices but from the edges.

Hierarchies. Attribute-based hierarchies are obtained by filtering the min-tree Q_{CT} of the MST, and computing the full hierarchy amounts to reweighting the MST. Intuitively, it is as if colored water fills up the branches of Q_{CT}, a merging of two components taking place when two different colors meet. The speed of the filling is controled by the attribute, *i.e.*, the branch is completely filled (*i.e.*, the branch is cut) when the amount of water is exactly equal to the attribute of the corresponding node of Q_{CT}. Some nodes exist in Q_{BT} but not in Q_{CT}. Those nodes correspond to an information not present in Q_{CT}: the order of processing, useful when a choice has to be made. A complete illustrative example explaining the process is detailed in Fig. 3. Computing the correct attribute is done thanks to the function `getAttribute`, that computes from `attributeComp` the attribute at the time of the merging. It consists in taking the highest attribute of all the children with the same original weight. The final weight of the MST can then be obtained by taking the lowest attribute of the two children of the node corresponding to the edge of the MST in Q_{BT}; this is done by `ComputeMergeAttributeMST`, the complexity of the whole post-processing being in $O(|V|)$. Once the re-weighted MST is computed, the hierarchical tree can be obtained by re-applying the algorithms of this paper. As seen in [1], the resulting hierarchy can be either a hierarchy of watershed cuts or a constrained-connectivity hierarchy.

Function getAttribute(n)

```
Data: A node $n$ of $Q_{BT}$
Result: The attribute at the time of the merging
if (n is the root) or (weightNode(parent[n]) $\neq$ weightNode(n)) then
    for all c children of n do getAttribute(c);
    attribute[n] := attributeComp[n];
else
    max:=0;
    for all children c of n do
        v:=getAttribute(c);
        if $v > max$ then max := v;
    attribute[n] := max;
return attribute[n];
```



(a) Q_{BT}

(b) Q_{CT}

Fig. 3. Hierarchical trees Q_{BT} and Q_{CT}. At the bottom of (a) and (b) is the (edge-weighted) graph of a map with the respective weight of each edge. Attributes (in this example, the surface of a node) can easily be computed on either Q_{BT} or Q_{CT}. In (a), the numbers (k, l) in parenthesis represents respectively k=`attributeComp`[n] and l=`getAttribute`(n). In (a) and (b), the numbers k in $n_i : k$ represents k=`attributeComp`[n]. When a flooding-by-attribute is performed on Q_{CT}, node n_1 disappears first at value 2, followed by node n_2 at value 3. When these the two corresponding branches are filled by water (cut from the tree), n_1 and n_2 are no longer minima. This is not the case for n_3, whose attribute is 6. When the corresponding branch is filled, n_3 stills marks a minimum, and stays so until the value 11, corresponding to node n_7. Thanks to the processing order embodied in Q_{BT}, `getAttribute` can compute the value at which a node disappears.

```
Procedure ComputeMergeAttributeMST
Data: $Q_{BT}$
Result: a reweighted MST $G$ corresponding to the attribute-based hierarchy
for any non-leaf node $n$ of $Q_{BT}$ do
    $a_1 := attribute[children[n].left]$;
    $a_2 := attribute[children[n].right]$;
    $G[getEdge(n)] := \min(a_1, a_2)$;
```

4 Conclusion

This paper has presented several elegant yet efficient algorithms for computing several morphological trees. At the heart of the processing is the minimum spanning tree, and in this paper we have proposed some variations on Kruskal algorithm. However, other approaches can be taken, and any other MST algorithm can be used first to produce a tree on which the algorithms of this paper can be applied. This would be needed in some situations, for example if the edges of the original graph do not fit in memory or if some parallel algorithm for minimum spanning tree is first needed. The unification theory provided in [1], together with the algorithms of this paper shed a new light on what has been done in mathematical morphology for a number of years, linking together some previously unrelated parts of the field.

Source code corresponding to this paper is available at
`http://www.esiee.fr/~info/sm/`.

Acknowledgements. This work received funding from the Agence Nationale de la Recherche, contract ANR-2010-BLAN-0205-03 and through "Programme d'Investissements d'Avenir" (LabEx BEZOUT n°ANR-10-LABX-58).

References

1. Cousty, J., Najman, L., Perret, B.: Constructive links between some morphological hierarchies on edge-weighted graphs. In: Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 86–97. Springer, Heidelberg (2013)
2. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE TIP 7(4), 555–570 (1998)
3. Nagao, M., Matsuyama, T., Ikeda, Y.: Region extraction and shape analysis in aerial photographs. CGIP 10(3), 195–223 (1979)
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. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE TIP 9(4), 561–576 (2000)
6. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. IEEE PAMI 18(12), 1163–1173 (1996)
7. Morris, O.J., de Lee, M.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)
8. 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)
9. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle. IEEE PAMI 31(8), 1362–1374 (2009)
10. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE PAMI 30(7), 1132–1145 (2008)
11. Meyer, F., Najman, L.: Segmentation, minimum spanning tree and hierarchies. In: Najman, L., Talbot, H. (eds.) Mathematical Morphology: From Theory to Application, pp. 229–261. ISTE-Wiley, London (2010)
12. Najman, L.: On the equivalence between hierarchical segmentations and ultrametric watersheds. JMIV 40(3), 231–247 (2011) arXiv:1002.1887v2
13. Cousty, J., Najman, L.: Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 272–283. Springer, Heidelberg (2011)
14. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. In: Proceedings of the AMS, vol. 7, pp. 48–50 (February 1956)
15. Tarjan, R.: Efficiency of a good but not linear set union algorithm. Journal of the ACM 22, 215–225 (1975)
16. Meyer, F.: Minimum spanning forests for morphological segmentation. In: ISMM, pp. 77–84 (September 1994)

Optima on Hierarchies of Partitions

Jean Serra and Bangalore Ravi Kiran

Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, A3SI, ESIEE
{j.serra,kiranr} @esiee.fr

Abstract. A new approach is proposed for finding optimal cuts in hierarchies of partitions by energy minimization. It rests on the notion of h-increasingness, allows to find best(optimal) cuts in one pass, and to obtain nice "climbing" scale space operators. The ways to construct h-increasing energies, and to combine them are studied, and illustrated by two examples on color and on textures.

1 Introduction

A hierarchy, or pyramid, of image segmentations is as a series of progressive simplified versions of an initial image, which result in increasing partitions of the space. We propose to reduce such pyramids to some *best* or optimal segmentation [1]. We shall not focus on the methods for obtaining the pyramids, and consider rather the whole hierarchies as starting points. Now, a multi-scale image description can rarely be considered as an end in itself. It often requires to be completed by some energy function ω that allows us to formalize optima. More precisely, three pieces of information interact, namely a pyramid H of partitions of the space E (possibly segmentations of an input image), some *energetic* function f on E which may be the initial image, or another one, and an energy ω over the partial partitions $\mathcal{D}(E)$, and which depends of f. Three questions arise then, namely:

1. Given a hierarchy H of partitions and a energy ω, how to obtain a new partition that minimizes ω, without getting bogged down in the combinatorial complexity?
2. How a family $\{\omega^j, j \in J\}$ can be a scale space operator, and generate a sequence of optimal partitions that increase with j?
3. Most of the segmentations involve several features (colour, shape, size, etc.), that one can handle with different energies ω. How to combine them, according to which grammar?

These questions, which are treated below in this order, have been taken up by several authors, for many years, and by various methods. Some of them [10], [6] simplify the combinatorial complexity by assuming that the energy of a partition

[1] This work received funding from the Agence Nationale de la Recherche through contract ANR-2010-BLAN-0205-03 KIDIKO.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 147–158, 2013.

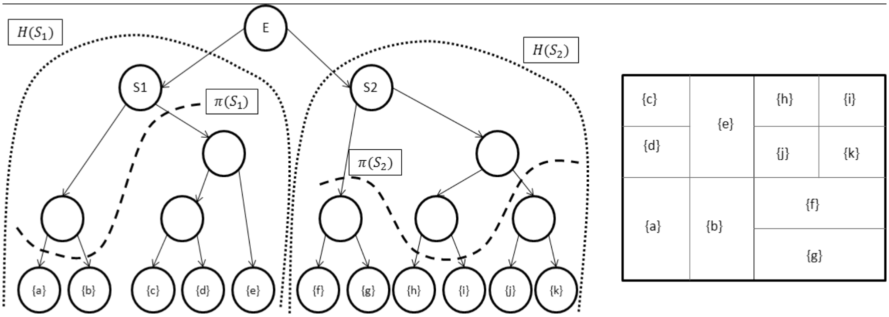


Fig. 1. Example of a dendrogram, with the corresponding partitions

equals the sum of the energies of its classes, which permits the treat the above questions 1) and 3). However, one can wonder whether additivity is the very underlying cause, since alternative approaches [12] replace the sums by suprema and obtain similar properties. In fact, the corner stone for a method which aims to solve questions 1) to 3) is the notion of $h-$increasingness introduced in [11], which encompasses all above particular assumptions. This following sections demonstrate this.

2 Hierarchies and Cuts (Reminder)

2.1 Hierarchies of Partial Partitions

We denote by E a 2-D topological space, such as a subset of $\mathbb{R}^2$ or $\mathbb{Z}^2$. A partition $\pi(S)$ associated with a set $S \in \mathcal{P}(E)$ is called *partial partition* of E of support S [9]. The partial partition of S in the single class S is denoted by $\{S\}$. The family of all partial partitions of set E is denoted by $\mathcal{D}(E)$, or simply by $\mathcal{D}$. A hierarchy H is a chain of partitions π_i, i.e.

$$H = \{\pi_i, 0 \leq i \leq n \mid i \leq k \leq n \Rightarrow \pi_i \leq \pi_k\}, \tag{1}$$

where π_n is the partition $\{E\}$ of E in a single class, called the *root*. The classes of the finest partition π_0 are called the *leaves*. The intermediary classes are also called *nodes*. The number of leaves is supposed to be finite, so that n, the number of levels is also finite.

Let $S_i(x)$ be the class of partition π_i of H at point $x \in E$. Expression (1) means that at each point $x \in E$ the family $\{S_i(x), x \in E, 0 \leq i \leq n\}$ of those classes $S_i(x)$ that contain x forms a finite chain of nested elements from the leaf $S_0(x)$ to E.

According to a classical result, a family $\{S_i(x), x \in E, 0 \leq i \leq n\}$ of indexed sets generates the classes of a hierarchy iff $i \leq j$ and $x, y \in E$ implies

$$S_i(x) \subseteq S_j(y) \quad \text{or} \quad S_i(x) \supseteq S_j(y) \quad \text{or } S_i(x) \cap S_j(y) = \varnothing. \tag{2}$$

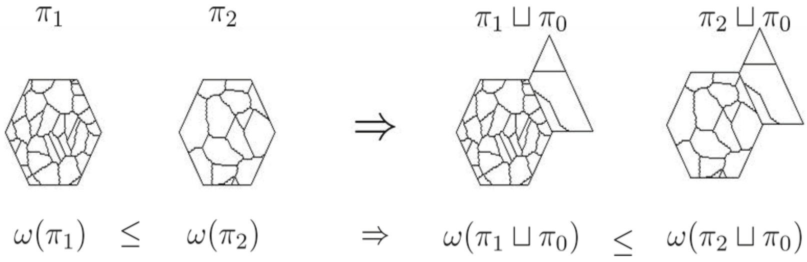

π_1
π_2
$\pi_1 \sqcup \pi_0$
$\pi_2 \sqcup \pi_0$
$\Rightarrow$
$\omega(\pi_1) \leq \omega(\pi_2) \Rightarrow \omega(\pi_1 \sqcup \pi_0) \leq \omega(\pi_2 \sqcup \pi_0)$

Fig. 2. Hierachical increasingness

The partitions of a hierarchy may be represented by their classes, or by the saliency map of the edges, or again by a dendrogram where each node of bifurcation is a class S, as depicted in Figure 1. The classes of π_{i-1} at level $i-1$ which are included in class $S_i(x)$ are said to be *the sons* of $S_i(x)$. Denote by $\mathcal{S}(H)$ the set of all classes S of all partitions involved in H. Clearly, the descendants of each S form in turn a hierarchy $H(S)$ of root S, which is included in the complete hierarchy $H = H(E)$.

2.2 Cuts in a Hierarchy

Any partition π of E whose classes are taken in $\mathcal{S}$ defines a *cut* π in a hierarchy H. The set of all cuts of E is denoted by $\Pi(E) = \Pi$. Every "horizontal" section $\pi_i(H)$ at level i is obviously a cut, but several levels can cooperate in a same cut, such as $\pi(S_1)$ and $\pi(S_2)$, drawn with thick dotted lines in Figure 1. Similarly, the partition $\pi(S_1) \sqcup \pi(S_2)$ generates a cut of $H(E)$. The symbol $\sqcup$ is used here for expressing that groups of classes are concatenated, i.e.

$$S = S_1 \sqcup S_2 \Leftrightarrow S = S_1 \cup S_2 \text{ and } S_1 \cap S_2 = \varnothing$$

One can also define cuts inside any sub-hierarchy $H(S)$ of summit S, and similarly, $\Pi(S)$ stands for the family of all cuts of $H(S)$.

3 Optimization and Hierarchical Increasingness

3.1 Energies and Optimization

The family of all p.p. of the leaves is denoted by $\mathcal{D}$. An energy on $\mathcal{D}$ is a non negative function $\omega : \mathcal{D} \rightarrow [0, \infty]$. In the following, $\mathcal{D}$ will be provided with several energies ω, which may satisfy the two following axioms (for the existence of nice optimal cuts, and for their unicity):

i) ω is *h-increasing*, i.e.

$$\omega(\pi_1) \leq \omega(\pi_2) \quad \Rightarrow \quad \omega(\pi_1 \sqcup \pi_0) \leq \omega(\pi_2 \sqcup \pi_0). \tag{3}$$

where π_1 and π_2 are two p.p. of same support S, and π_0 a p.p. of support S_0 disjoint of S [11]. The geometrical meaning of Rel.(3) is depicted in Figure 2.

ii) ω is *singular*, when the energy $\omega(\{S\})$ of class S is differs from that of any p.p. of S, i.e.

$$\pi(S) \text{ p.p. of } \{S\} \quad \Rightarrow \quad \omega(\{S\}) \neq \omega(\pi(S)). \tag{4}$$

The optimization problem involves three entities:

1. A pyramid H of partitions of E which segment an input image,
2. An energy ω on the family $\mathcal{D}(E)$ of all partial partitions of E,
3. An "energetic" function f on E which may be the initial image, or another one, which parametrizes energy ω.

These three pieces of information are independent, and aim to determine the cuts that minimizes ω, i.e. such that $\omega(\pi^*) = \inf\{\omega(\pi) \mid \pi \in \Pi(E)\}$. They are called below the *optimal cuts*.

3.2 Optimal Cut Characterization

Though the hierarchies are discrete, the number of their possible cuts explodes combinatorially: a small hierarchy of 200 leaves and 10 levels generates billions of cuts! How to find out the best one? By means of which vital lead? The h-increasingness (3) turns out to be too demanding and too general, since it does not take into account that we are dealing with hierarchies. We are thus lead to replace it by the following weaker but more adapted version of h-increasingness. We have to introduce the set $\mathcal{H}$ of all finite hierarchies of partitions of E.

Definition 1. *An energy ω on $\mathcal{D}(E)$ is* weakly h-increasing *when for any hierarchy $H \in \mathcal{H}$, any disjoint nodes S and S_0 of H, and any partition π_0 of S_0, we have*

$$\omega(\pi^*) = \inf\{\omega(\pi),\ \pi \in \Pi(S)\} \Rightarrow\ \omega(\pi^* \sqcup \pi_0) \leq \inf\{\omega(\pi \sqcup \pi_0),\ \pi \in \Pi(S)\} \tag{5}$$

where $\Pi(S)$ stands for the finite set of all partitions of node S involved in hierarchy H.

Clearly, h-increasingness implies weak h-increasingness, i.e. Rel.(3) ⇒ Rel.(5). More precisely, Rel.(3) has been weakened just enough to obtain the theorem of optimal cut working in both senses. Indeed, we now have

Theorem 1. *Let $H \in \mathcal{H}$ be a finite hierarchy, and ω an energy on $\mathcal{D}(E)$, and S be a node of H of sons $T_1..T_p$. If $\pi_1^*, ..\pi_p^*$ are cuts of optimal energies of $T_1..T_p$ respectively, then*

$$\pi_1^* \sqcup \pi_2^* .. \sqcup \pi_p^*, \tag{6}$$

is an optimal cut of $\Pi(S) \setminus \{S\}$, for any $H \in \mathcal{H}$ and any $T_1..T_p$ in H, if and only if ω is weakly h-increasing.

Proof for theorem 1 is given in [7].

When the h-increasing energy ω is also singular, then theorem 1 leads to the following key consequence

Corollary 1. *Let ω be h-increasing and singular energy. Then for any $H \in \mathcal{H}$ and any node S of H with p sons $T_1..T_p$ of optimal cuts $\pi_1^*, ..\pi_p^*$, there exists a unique optimal cut of the sub-hierarchy of root S. It is either the cut $\pi_1^* \sqcup \pi_2^*..\sqcup \pi_p^*$, or the one class partition $\{S\}$ itself:*

$$\omega(\pi^*(S)) = \min\{\omega(\{S\}), \omega(\pi_1^* \sqcup \pi_2^*.. \sqcup \pi_p^*)\} \tag{7}$$

Corollary 1 is essential. It governs the choices of models for energies, and their implementations:

Firstly, the obtained optimal cut $\pi^*(E)$ is indeed globally less energetic than any other cut in H, but, moreover, each class $S \in \pi^*(E)$ is less energetic than any p.p. of S into classes of H, and also less energetic than any p.p. composed of classes of H and containing S. This is a strong property of regional minimum.

Secondly, the condition (3) of h-increasingness for an energy being a notion independent of any hierarchy, one can use a different ω for each of the n levels of hierarchy H.

Thirdly, dealing with h-increasingness is sufficient. Fortunately so, because it is incomparably easier to check the h-increasingness of an energy than its possible weak h-increasingness.

Fourthly, the optimal cut coincides with the min-cut in the sense of the max-flow methods on graphs when one takes for source the set of leaves, and for sink the whole space E.

Finally one can always impose unicity, for example by taking systematically $\{S\}$ and not $\pi^*(S)$ in case of equal energies at node S. This technique of choice makes h-increasingness and sigularity compatible [7].

3.3 Generation of h-Increasing Energies

An easy way to obtain a h-increasing energy consists in defining it, firstly, over all sets $S \in \mathcal{P}(E)$, considered as one class partial partitions $\{S\}$, and then in extending it to partial partitions by some law of composition. Then, the h-increasingness is introduced by the law of composition, and not by $\omega[\mathcal{P}(E)]$. The first two modes of composition which come to mind are, of course, addition and supremum, and indeed we can state

Proposition 1. *Let E be a set and $\omega : \mathcal{P}(E) \to \mathbb{R}^+$ an arbitrary energy defined on $\mathcal{P}(E)$, and let $\pi \in \mathcal{D}(E)$ be a partial partition of classes $\{S_i, 1 \leq i \leq n\}$. Then the two extensions of ω to the partial partitions $\mathcal{D}(E)$ by addition and by supremum*

$$\omega(\pi) = \sum\{\omega(S_i), 1 \leq i \leq n\} \quad \text{and} \quad \omega(\pi) = \vee\{\omega(S_i), 1 \leq i \leq n\}$$

are h-increasing energies. Moreover, if $\{\alpha_j, j \in J\}$ stands for a family of non negative weights, then the weighed sum $\sum\alpha_j\omega_j$ and supremum $\bigvee\alpha_j\omega_j$ of h-increasing energies turn out to be $h-$increasing.

(Easy proof). A number of other laws are compatible with h-increasingness. One can also make ω depend on more than one class, on the proximity of the edges, on another hierarchy, etc..

4 Scale Increasingness and Climbing Energies

The usual energies are often given by finite sequences $\{\omega^{\lambda}, \lambda \in \Lambda\}$ that depend on a real positive index, or parameter, λ which takes p different values, $p < \infty$. Therefore, the processing of hierarchy H results in a sequence of p optimal cuts $\pi^{\lambda *}$, of labels $\lambda \in \Lambda$. *A priori*, the $\pi^{\lambda *}$ are not ordered, but if they were, then we should obtain a nice progressive simplification of the optimal cuts. For getting it, we need to combine h-increasingness with the supplementary axiom (8) of *scale increasingness*, which results in the following *climbing energies*.

Definition 2. *We call* climbing energy *any family* $\{\omega^{\lambda}, \lambda \in \Lambda\}$ *of energies over* $\mathcal{D}$ *which satisfies the three following axioms, valid for each* ω^{λ} *and for all* $\pi \in \Pi(S)$, $S \in \mathcal{S}$.
i) each $\omega^{\lambda}, \lambda \in \Lambda$, *is* h-increasing,
ii) each $\omega^{\lambda}, \lambda \in \Lambda$, *is* singular*,*
iii) the $\{\omega^{\lambda}, \lambda \in \Lambda\}$ *are* scale increasing, *i.e. for* $\lambda \leq \mu$ *, each support* $S \in \mathcal{S}$ *and each partition* $\pi \in \Pi(S)$*,* we have

$$\lambda \leq \mu \text{ and } \omega^{\lambda}(S) \leq \omega^{\lambda}(\pi(S)) \Rightarrow \omega^{\mu}(S) \leq \omega^{\mu}(\pi(S)), \quad \pi \in \Pi(S), \quad S \in \mathcal{S}. \tag{8}$$

Axiom *i)* compares the same energy at two different levels, whereas axiom *iii)* compares two different energies at the same level. The relation (8) means that, as λ increases, the ω^{λ} preserve the sense of energetic differences between the nodes of hierarchy H and their partial partitions. In particular when ω_0 is h-increasing and singular, and when family $\{\omega^{\lambda}, \lambda \in \Lambda\}$ is climbing, then the two families $\{\lambda\omega\}$, and $\{\omega^{\lambda} + \omega_0, \lambda \in \Lambda\}$ are climbing.

4.1 Ordering and Computation of the Optimal Cuts

The climbing energies satisfy the very nice property to order the optimal cuts with respect to the parameter λ [7]:

Theorem 2. *Let* $\{\omega^{\lambda}, \lambda \in \Lambda\}$ *be a family of climbing energies, and let* $\pi^{\lambda *}$ *(resp.* $\pi^{\mu *}$*) denote the optimal cut of hierarchy* H *according to the energy* ω^{λ} *(resp.* ω^{μ}*). Then the family* $\{\pi^{\lambda *},\lambda \in \Lambda\}$ *of the optimal cuts generates a hierarchy* H^* *of partitions, i.e.*

$$\lambda \leq \mu \quad \Rightarrow \quad \pi^{\lambda *} \leq \pi^{\mu *}, \qquad \lambda, \mu \in \Lambda. \tag{9}$$

Computationally, the h-increasing condition (3) allows us to reach the optimal cut in one ascending pass, by the following Guigues'algorithm [6]:

- Scan in one pass all nodes of H in ascending lexicographic order.

- Determine at each node S a temporary optimal cut of H by comparing the energy of S to that of the concatenation of the temporary optimal cuts of the (already scanned) sons T_k of S.

In addition, the scale increasingness allows us to obtain the whole family of the optimal cuts in one ascending pass followed by a descending one [7] [6].

We will now review two families of climbing energies. The first one focuses on the additivity of the classes and the second on their supremum.

5 Additive Energies

The additive h-increasing mode was introduced and studied by L. Guigues et Al. under the name of *separable energies* [6], for partitions with connected classes. For the aim of scale increasingness, the energy $\omega^{\lambda}(S)$ is written as a linear function of λ:

$$\omega^{\lambda}(S) = \omega_{\varphi}(S) + \lambda\omega_{\partial}(S) \qquad S \in \mathcal{S}. \tag{10}$$

The additive family $\{\omega_{\varphi} + \lambda\omega_{\partial}\}$ is climbing iff the term ω_{∂} is sub-additive for union, i.e.

$$\omega_{\partial}(\bigcup_{1\leq u\leq q} T_u) \leq \sum_{1\leq u\leq q} \omega_{\partial}(T_u) \tag{11}$$

The climbing family $\{\omega^{\lambda}\}$ of Rel. (10) admits a nice lagragian interpretation if we view the term ω_{∂} as a constraint on the functional ω_{φ} to minimize. According to Lagrange formalism, given one constraint ω_{∂}, the optimum is reached by means of a system of partial derivatives. Now remarkably the current approach replaces that by a unique *climbing.* As the term $\omega_{\partial}(\pi)$ decreases as λ^j increases, we can climb the pyramid of the optimal cuts and stop (thus optimal λ) when the constraint is satisfied.

The most popular climbing additive energy was proposed by Mumford and Shah [8] and evolved under various forms. Let $\pi(S)$ be the partition of a summit S into its q sons $\{T_u, 1 \leq u \leq q\}$ i.e. $\pi(S) = T_1 \sqcup ..T_u.. \sqcup T_q$. The energy ω_{φ}, called *fidelity term*, sums up the quadratic differences between f and its average $m(T_u)$ in the various T_u, and the energy ω_{∂}, called *regularity term* weights by λ^j the lengths ∂T^i of the frontiers of all T_u, i.e.

$$\omega^j(\pi(S)) = \sum_{1\leq u\leq q} \int_{x\in T^u} \parallel f(x) - m(T_u) \parallel^2 + \lambda^j \sum_{1\leq u\leq q} (\partial T_u) = \omega_{\varphi}(\pi) + \lambda^j\omega_{\partial}(\pi) \tag{12}$$

where the weight λ^j is a numerical increasing function of the level number j.

The two terms of Rel.(12) are far from being the only possible ones. The second example of Section 8 below brings textures into play via inter-class variances. In [7], the convexity of S is introduced by comparing the positive and negative curvatures of ∂S. In [13], the quadratic differences of ω_{φ}, in Rel.(12), are replaced by two intergrals of f in the outer and inner parts of the dilate $\partial S \oplus B$ of ∂S by a disc B, etc..

6 Sup-generated Energies

Just as the sum-generated ones, the $\vee$-generated energies on the partial partitions are defined from an energy ω on $\mathcal{P}(E)$, followed by a law of composition, which is now the supremum

$$\omega(\pi) = \omega(T_1 \sqcup ... \sqcup T_n) = \vee\{\omega(T_i)\}. \tag{13}$$

6.1 Binary $\vee$-Generated Energies

These energies are proposed by P.Soille [12], with several variants. For example, a numerical function f is associated with the hierarchy H. The range of variation $\delta(S) = \max\{f(x), x \in S\} - \min\{f(x), x \in S\}$ of f inside set S defines the h-increasing binary energy $\omega^k(\langle S \rangle) = 0$ when $\delta(S) \leq k$, and $\omega^k(\langle S \rangle) = 1$ when not. The energy ω^k is extended to partial partitions by supremum, so that the class of the optimal cut at point $x \in E$ is the larger class of H whose range of variation is $\leq j$. When the energy ω^k of a father equals that of its sons, one keeps the father when $\omega^k = 0$, and the sons when not.

6.2 Ordered Energies

When they are not binary, some $\vee$-generated energies are presented via an ordering condition. As previously, an energy is still associated with each subset S of E. The axiom (3) of $h-$increasingness does not require we know the energy of all partial partitions. In particular, when the comparison of the partial partitions π_1 and π_2 reduces to that of their classes, then a law of composition becomes useless.

Definition 3. *An energy ω on $\mathcal{D}$ is said to be ordered when for all pairs π_1, $\pi_2 \in \mathcal{D}$ we have $\omega(\pi_1) \leq \omega(\pi_2)$ iff*

-Both π_1 and π_2 admit the same support Supp,
-For all points $x \in$ Supp, for classes $S_1(x)$ and $S_2(x)$ in π_1 and π_2 respectively, the inequality $\omega[\{S_1(x)\}] \leq \omega[\{S_2(x)\}]$ holds.

An ordered energy ω is always h-increasing. When S is the support of the partition $\pi = \sqcup T_i$, then $\omega(S) \leq \omega(\pi)$ iff $\omega(S) \leq \vee\omega(T_i)$, and we find again the $\vee$-composition.

Here is an example of ordered energy due to H.G.Akcay and S. Aksoy [1] who study airborne multi-band images and introduce (up to a small change) $\mu(S)$ =Area $(S) \times$ (mean of all standard deviations of all bands in S).They work with energy *maximization.* Allocate a non negative measure $\mu(S)$ to each node of a hierarchy H, where μ takes its values in a partially ordered set M, such as a color space. The energy ω is ordered by the two conditions

$$\omega(S) \leq \omega(S') \Leftrightarrow S \supseteq S' \text{ and } \mu(S) \geq \mu(S') \quad S\ ,\ S' \in \mathcal{P}(E),\ \mu \in M. \tag{14}$$

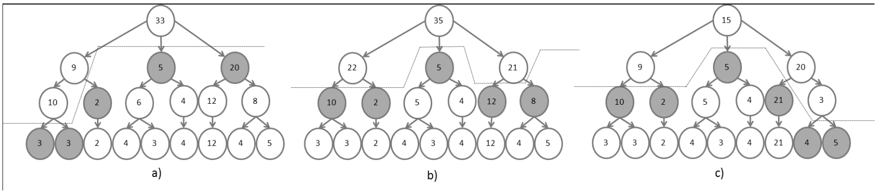


Fig. 3. Comparison of three laws of composition, a) by addition, b) by supremum, c) by ordering. The energies ω are indicated in the discs.

The node S^* of the optimal cut at point x is the highest more energetic than all its descendants. The best cut π^* is obtained in one pass, by Guigues' algorithm [6].

Figure 6.2 summarizes the three major laws of composition. In a) the additive mode chooses the father S, when $\omega(S) \leq \sum \omega(T_j)$. In b) the mode by supremum chooses the S, when $\omega(S) \leq \vee\ \omega(T_j)$. Finally, in c) one takes the largest node which is more energetic than all its descendants(maximization of ω).

6.3 Composition of $\vee$-Generated Energies

Though the weighted supremeum of $\vee$-generated energies is h-increasing (eqn.13), the infimum is not. In practice, this half-result is nevertheless useful, since the $\vee$, paradoxically, expresses the intersection of criteria. For example, when the function f to optimize is colour, one can take for energies:

- $\omega_1(S) = 0$ when $\delta Lum(S) < k_1$, and $\omega_1(S) = 1$ when not,
- $\omega_2(S) = 0$ when $\delta Sat(S) < k_2$, and $\omega_2(S) = 1$ when not.

Then the h-increasing energy $\omega_1(S) \vee \omega_2(S) = 0$ when S is constant enough for both luminance and saturation.

7 Partial Optimizations

Covering the whole space with some optimal partition is not always an aim. Some studies require doing it, but in others ones the regions of interest are limited, and clearly marked out by the context. Moreover, the leave partition often includes a good many classes due to noise. And thirdly, the hierarchies generated by connected filters may comprise a large number of singleton classes. For example, Figure 4 b) and c) depict the flat zones obtained by an alternating filter by reconstruction acting on the 25098 image a). All black pixels indicate the singleton flat zones. When climbing the hierarchy, most of these point classes are covered by extended classes, which are more significant. Therefore we can just ignore the singletons when the classes of H are given an energy.

In other situations, some classes may be considered as non relevant because they are too small, or too large, or too far from the zone of interest, or of a

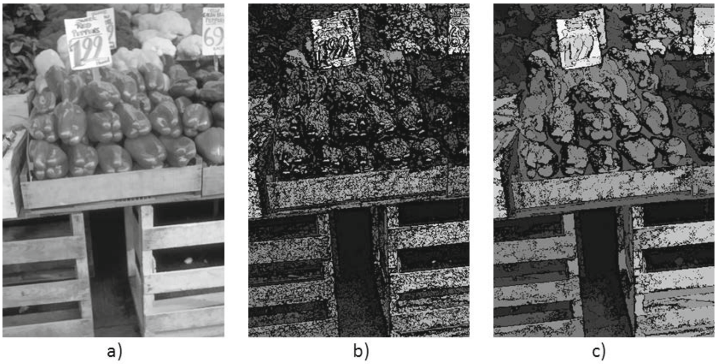


Fig. 4. a) image 25098 from Berkeley database b) and c) alternating filters of a), of sizes 1 and 5

non wanted hue, etc...In all cases, they are clearly identified, so that some label can indicate that they have not to intervene when computing the optimal cut. Denote by $\mathcal{W}(E) \subseteq \mathcal{P}(E)$ the set of all these undesirable classes. The energies ω must satisfy the condition that, for all families $\{S_i\} \subseteq \mathcal{P}(E)$ and all families $\{W_j\} \in \mathcal{W}(E)$ such that $(\cup_i S_i) \cap (\cup_j W_j) = \varnothing$, we have

$$\omega((\sqcup_i S_i) \sqcup (\sqcup_j W_j)) = \omega(\sqcup_i S_i).$$

The energy of the partial partition of classes $\{S_i\}$ must not change when outside $\{W_j\}$ classes are added. It means that $\omega(W) = 0$ when the law of composition invoved in ω is the sum or the supremum, and that $\omega(W) = \infty$ when it is the infimum. When ω is h-increasing, the computation of the optimal cut is unchanged, but now results in a partition which may contain W classes.

8 Two Examples

We now develop two examples of additive energies which aim to show how the choice of the energies governs the extraction of specific features (color, textures). We start from the energy proposed in [10], and change it by adding new terms. A hierarchy H of Uppsala ducks has been obtained by previous segmentations of the luminance $l = (r + g + b)/3$ based on [5]. We want to find the best cut for a compression rate of 20. In each class S of H, the simplification consists in replacing the function f by its colour mean (mean over all 3 channels) $\overline{l}(S) = \frac{\Sigma_{x \in S} I(x)}{card(S)}$. In a first experiment the energy $\omega_{lum}(S)$ has for fidelity term $\omega_{\varphi}(S)$ the quadratic error, while the regularity term $\omega_{\partial}(S)$ is the coding cost of class S, by taking 2 bits for each frontier element, and 24 bits assigned to code $m(S)$:

$$\omega_{lum}(S) = \sum_{x \in S} \parallel l(x) - \overline{l}(S) \parallel^2 + \lambda(24 + \mid \partial S \mid) \tag{15}$$

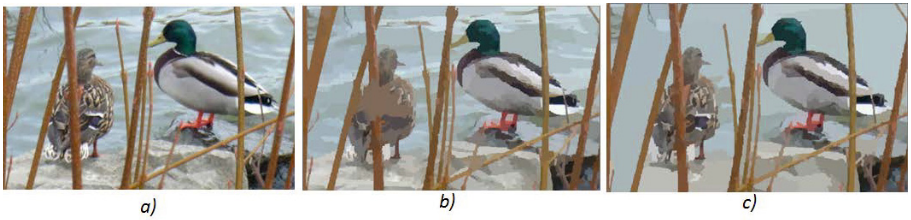


Fig. 5. a) Initital Uppsala ducks; Optimal cuts b) by Luminance c) by Chrominance

The cost $\omega_\partial(\pi^*)$ of the best cut decreases as λ^j increases, therefore we can climb the pyramid of the best cuts and stop when $\omega_\partial(\pi) \simeq$ pixels number/20. It results in Fig. 5b, where we see that the female duck is not nicely simplified. In the second experiment, we just replace the luminance l, in Rel.(15) by the chrominance c, i.e. by the projection of (r, g, b) on the plane orthogonal to the main axis $(1, 1, 1)$. This simplifies the image while keeping partitions which minimize the variance of the chrominance vector c. The new optimal cut is depicted in Fig 5c. Both ducks are better separated from the foreground of the herbs, and from the background of the river.

$$\omega_{textures}(S) = \omega_{chrom}(S) + \sum_{S\prime \in sons(S)} \frac{K}{\sigma^2(Area(S'))} \qquad (16)$$

The second example addresses to the recognition of textures in the trees, in the walls and in the water of the Uppsala river of Fig. 6a. The new energy $\omega_{textures}$ of Rel.(16) keeps the first two terms of the previous experiment.

The third term of (16) introduces the textures via the variances of the sons. It decreases more drastically when the areas of the sons of S have similar sizes. Fig. 6b and c depict the best cuts for two values of the parameters. Intuitively, texture features are formulated into this multi-scale framework where the optimal scale parameter combines the effect of chrominance and structure of texture into one global energy function, thus showing the flexibility of the framework.

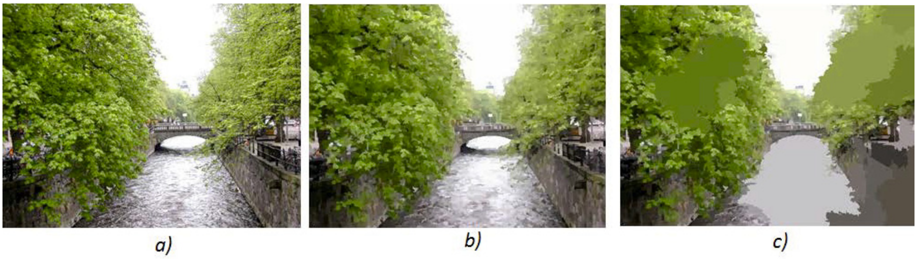


Fig. 6. a) Uppsala river; b) and c) best cut of a) according to energy (16), for $\lambda = 100$, and for $K = 10^{12}$ (in b)), and $K = 10^{14}$ (in c))

9 Conclusion

The primary contributions of this theoretical paper were:

- Hierarchical optimizations based on h-increasingness, while also giving the conditions for general classes to be h-increasing.
- Defining non negative global *climbing energies* that allow to perform sequences of increasing optimizations over a hierarchy of segmentations.
- Demonstrating how to formulate multiple constraint functions over the image space in order to lead to different optimal segmentations. Two examples, one with colour image segmentation and one with texture enhancement were shown.

Acknowledgements. We are thankful to Professor Fernand Meyer for his helpful comments on the converse of theorem 1.

References

1. Akcay, H.G., Aksoy, S.: Automatic detection of geospatial objects using multiple hierarchical segmentations. IEEE T. Geoscience and Remote Sensing 46, 2097–2111 (2008)
2. Angulo, J., Serra, J.: Modeling and segmentation of colour images in polar representations. Image and Vision Computing 25, 475–495 (2007)
3. Arbelez, P., Cohen, L.: Constrained Image Segmentation from Hierarchical Boundaries. In: CVPR (2008)
4. Cardelino, J., Caselles, V.: Bertalm ´ ıo, M., Randall, G.: A contrario hierarchical image segmentation. In: IEEE ICIP 2009, Cairo, Egypt (2009)
5. Cousty, J., Najman, L.: Incremental Algorithm for Hierarchical Minimum Spanning Forests and Saliency of Watershed Cuts. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 272–283. Springer, Heidelberg (2011)
6. Guigues, L., Cocquerez, J.P., Le Men, H.: Scale-Sets Image Analysis. Int. Journal of Computer Vision 68(3), 289–317 (2006)
7. Kiran, B.R., Serra, J.: Global-local optimizations on hierarchies of segmentations. Special Issue on Computer Vision Applying PR, Pattern Recognition Letters (2013)
8. Mumford, D., Shah, J.: Boundary Detection by Minimizing Functionals. In: Ulmann, S., Richards, W. (eds.) Image Understanding (1988)
9. Ronse, C.: Partial partitions, partial connections and connective segmentation. Journal of Mathematical Imaging and Vision 32, 97–125 (2008)
10. Salembier, P., Garrido, L.: Binary Partition Tree as an Efficient Representation for Image Processing, Segmentation, and Information Retrieval. IEEE Trans. on Image Processing 9(4), 561–576 (2000)
11. Serra, J.: Hierarchies and Optima. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) DGCI 2011. LNCS, vol. 6607, pp. 35–46. Springer, Heidelberg (2011)
12. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Transactions on PAMI 30, 1132–1145 (2008)
13. Xu, Y., Géraud, T., Najman, L.: Morphological Filtering in Shape Spaces: Applications using Tree-Based Image Representations. In: ICPR 2012 (2012)

Semi-connections and Hierarchies*

Olena Tankyevych[1], Hugues Talbot[2], and Nicolas Passat[3]

[1] Université Paris-Est, LISSI, Créteil, France
[2] Université Paris-Est, LIGM, ESIEE-Paris, France
[3] Université de Reims, CReSTIC, France

Abstract. Connectivity is the basis of several methodological concepts in mathematical morphology. In graph-based approaches, the notion of connectivity can be derived from the notion of adjacency. In this preliminary work, we investigate the effects of relaxing the symmetry property of adjacency. In particular, we observe the consequences on the induced connected components, that are no longer organised as partitions but as covers, and on the hierarchies that are obtained from such components. These hierarchies can extend data structures such as component-trees and partition-trees, and the associated filtering and segmentation paradigms, leading to improved image processing tools.

Keywords: connectivity, cover hierarchy, connected operators, filtering.

1 Introduction

Connectivity plays a crucial role in the definition of mathematical morphology. Intuitively, the notion of connectivity serves to decide whether a set is either in one piece, or split into several. This notion has been widely studied [2], from axiomatic definitions [21] to variants such as constrained connectivity [24], second-generation connectivity [15,22,7,11], and hyperconnections [14].

Practically, connectivity in discrete image processing is often handled in graph-based frameworks, via the notion of adjacency [16]. In this context, connectivity has led to the development of data structures based on the partition of discrete spaces, and further on partition hierarchies. Examples of such hierarchies are component-trees [19,8], partition-trees [18,24], or hierarchical watershed [4,9,10]. They can also derive from connectivity hierarchies, leading to partition-trees based, *e.g.*, on fuzzy connectedness [17,1] or second-generation connectivity.

From an applicative point of view, all these concepts have been involved in the development of connected operators [20,23], devoted in particular to image processing tasks such as filtering or segmentation. In this article, we present a preliminary study on the effects of relaxing the symmetry hypothesis, actually required to define adjacency relations (Secs. 2–3). We observe that partitions then become *covers*, which leads us to define cover hierarchies instead of partition hierarchies. We prove however that such hierarchies can still be handled as (enriched) tree structures (Sec. 4). This framework generalises standard notions such as component-trees or partition-trees (Sec. 5), and provides solutions for performing more accurate antiextensive filtering tasks (Sec. 6).

* The research leading to these results has received funding from the French *Agence Nationale de la Recherche* (Grant Agreement ANR-2010-BLAN-0205).

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 159–170, 2013.

2 Background Notions: From Adjacency to Partition Hierarchies

We first recall basic definitions and properties related to the concept of adjacency and some induced notions, namely connectedness, partitions and partition hierarchies.

2.1 Adjacency and Connectedness

Let Ω be a nonempty finite set. Let $\rightleftharpoons$ be an adjacency (*i.e.*, irreflexive, symmetric, and binary) relation on Ω. (We recall that $\rightleftharpoons$ is a subset of $\Omega \times \Omega$.) If $x, y \in \Omega$ satisfy $x \rightleftharpoons y$ (and thus $y \rightleftharpoons x$), we say that x and y are adjacent.

Let $X \subseteq \Omega$ be a nonempty subset of Ω. Let $\rightleftarrows_X$ be the equivalence relation on X induced by the reflexive-transitive closure of the restriction $\rightleftharpoons_X$ of $\rightleftharpoons$ to X. If $x, y \in X$ satisfy $x \rightleftarrows_X y$, we say that x and y are connected (in X). In particular, the equivalence classes of X associated to the relation $\rightleftarrows_X$ are called the connected components of X, and the set of these connected components is noted $C_{\rightleftarrows}[X]$.

Remark 1. *These definitions are directly linked to classical notions on graphs, considered for the topological modelling of digital images, as introduced,* e.g.*, in [16]. In particular,* $(\Omega, \rightleftharpoons)$ *and* $(X, \rightleftharpoons_X)$ *are irreflexive (non-directed) graphs.*

2.2 Partition Hierarchies

The notion of connected component, associated to (any subsets of) $(X, \rightleftharpoons_X)$ is important in image analysis. Indeed, the partition $C_{\rightleftarrows}[X]$ associated to images defined on X, can be used for filtering and segmentation purpose, by considering approaches that rely on partition hierarchies in the framework of connected operators [20,23].

In this context, there exist two ways to refine $(X, \rightleftharpoons_X)$, to build partition hierarchies. The first way is to work on X, and to define subsets $Y \subseteq X$, *i.e.*, to progressively constrain the spatial part of $(X, \rightleftharpoons_X)$. Practically, defining $(Y, \rightleftharpoons_Y)$ such that $Y \subseteq X$, implies that $\rightleftharpoons_Y = (\rightleftharpoons_X \cap (Y \times Y))$. The second way is to work on $\rightleftharpoons_X$, and to define (symmetric) subrelations $\overset{\blacktriangle}{\rightleftharpoons}_X \subseteq \rightleftharpoons_X$, inducing connectedness relations $\overset{\blacktriangle}{\rightleftarrows}_X \subseteq \rightleftarrows_X$, *i.e.*, to progressively constrain the structural part of $(X, \rightleftharpoons_X)$.

Remark 2. *In the framework of graphs,* $(Y, \rightleftharpoons_Y)$ *is a subgraph of* $(X, \rightleftharpoons_X)$*, and* $(X, \overset{\blacktriangle}{\rightleftharpoons}_X)$ *is a partial graph of* $(X, \rightleftharpoons_X)$*.*

In both cases, we have the following property.

Property 3. *Let* $x \in Y$ *(resp.* $x \in X$*). Let* $C = C^x_{\rightleftarrows}(Y) \in C_{\rightleftarrows}[Y]$ *(resp.* $C = C^x_{\overset{\blacktriangle}{\rightleftarrows}}(X) \in C_{\overset{\blacktriangle}{\rightleftarrows}}[X]$*), and* $C^x_{\rightleftarrows}(X) \in C_{\rightleftarrows}[X]$ *be the unique connected components containing* x*. We have*

$$C \subseteq C^x_{\rightleftarrows}(X) \tag{1}$$

Moreover, for any $K \in C_{\rightleftarrows}[X]$*, we have*

$$\big(K \cap C \neq \varnothing\big) \Rightarrow \big(K = C^x_{\rightleftarrows}(X)\big) \tag{2}$$

By progressively refining either X into successive subsets, or $\rightleftharpoons_X$ into successive (symmetric) subrelations, we can then build partition hierarchies defined as *trees*[1].

Property 4. *Let $(X_i)_{i=0}^k$ (resp. $(\rightleftharpoons_i)_{i=0}^k$) $(k \geqslant 0)$ be such that $X_0 = X$ (resp. $\rightleftharpoons_0 = \rightleftharpoons_X$) and $X_{i+1} \subseteq X_i$ (resp. $\rightleftharpoons_{i+1} \subseteq \rightleftharpoons_i$) for all $i \in [\![0, k-1]\!]$. Let $(\bigcup_{i=0}^k C_{\rightleftarrows}[X_i], \sqsubseteq)$ (resp. $(\bigcup_{i=0}^k C_{\rightleftarrows_i}[X], \sqsubseteq)$) be the partially ordered multiset*[2] *where for all $K_\alpha \in C_{\rightleftarrows}[X_\alpha]$ (resp. $C_{\rightleftarrows_\alpha}[X]$) and $K_\beta \in C_{\rightleftarrows}[X_\beta]$ (resp. $C_{\rightleftarrows_\beta}[X]$) $(\alpha, \beta \in [\![0, k]\!])$, the order relation $\sqsubseteq$ extends $\subseteq$ as*

$$\big(K_\beta \sqsubseteq K_\alpha\big) \Leftrightarrow \Big(\big(K_\beta \subseteq K_\alpha\big) \wedge \big(\alpha \leqslant \beta\big)\Big) \tag{3}$$

For any $K, K' \in \bigcup_{i=0}^k C_{\rightleftarrows}[X_i]$ (resp. $\bigcup_{i=0}^k C_{\rightleftarrows_i}[X]$), we then have

$$\big(K \cap K' \neq \emptyset\big) \Rightarrow \big((K \sqsubseteq K') \vee (K' \sqsubseteq K)\big) \tag{4}$$

2.3 Partial and Total Partition Hierarchies

Both ways to refine $(X, \rightleftharpoons_X)$ lead to partition hierarchies, but they differ with respect to the nature of these partitions. Indeed, the structural refinement leads to total partitions, while the spatial refinement leads to partial partitions (as defined in [15]).

Property 5. *We have, for all $i \in [\![0, k-1]\!]$*

$$\begin{aligned} \bigcup C_{\rightleftarrows}[X_{i+1}] &\subseteq \bigcup C_{\rightleftarrows}[X_i] \\ \bigcup C_{\rightleftarrows_{i+1}}[X] &= \bigcup C_{\rightleftarrows_i}[X] \end{aligned} \tag{5}$$

Remark 6. *Typical examples of partial partition hierarchies are component-trees [19], where the successive $C_{\rightleftarrows}[X_i]$ are defined by considering the binary images obtained by thresholding a grey-level image $\mathcal{I} : X \to [\![0, k]\!]$. Typical examples of total partition hierarchies are (binary) partition-trees [18], where the successive $C_{\rightleftarrows_i}[X]$ are defined by progressively merging elementary parts of X, in a (multivalued) image $\mathcal{I} : X \to V$.*

3 Non-symmetry in Adjacency: Semi-adjacency

In Sec. 2, a crucial hypothesis was the symmetry of the adjacency relation $\rightleftharpoons_X$ defined on X. We now investigate the effects induced by the relaxation of this hypothesis.

3.1 Semi-adjacency

Adjacency is defined as a relation being both irreflexive and symmetric. By relaxing the symmetry hypothesis, the obtained relation may (most of the time) no longer be an adjacency. We then introduce a more general notion to handle that case.

Definition 7 (Semi-adjacency). *Let $\rightharpoonup$ be an irreflexive binary relation on Ω. Such a relation is called a* semi-adjacency *on Ω. If $x \rightharpoonup y$, we say that x is* semi-adjacent *to y.*

We recall that, similarly to $\rightleftharpoons$, the relation $\rightharpoonup$ is still a subset of $\Omega \times \Omega$. However, by opposition to $\rightleftharpoons$, we have $(x \rightharpoonup y) \nRightarrow (y \rightharpoonup x)$.

[1] Such trees are indeed *forests*, due to the non-necessary existence of a maximum (*i.e.*, a unique maximal element). For the sake of readability, the term tree is used by abuse of notation.

[2] It may happen that successive partitions contain some similar connected components.

3.2 Semi-connectedness

In the case of adjacency, the reflexive-transitive closure led to an equivalence relation that characterised the notion of connectedness. We follow here the same approach.

Definition 8 (Semi-connectedness). *Let $X \subseteq \Omega$ be a nonempty subset of Ω. We define the* semi-connectedness *relation $\rightarrow_X$ on X as the binary relation defined by the reflexive-transitive closure of the restriction $\rightharpoonup_X$ of $\rightharpoonup$ on X. If $x, y \in X$ satisfy $x \rightarrow_X y$, we say that x is* semi-connected *to y (in X).*

By opposition to $\rightleftarrows_X$, the relation $\rightarrow_X$ is not an equivalence relation, in general. (Note that a "semi-adjacency-based" notion of *stream* had also been considered in [3].)

Property 9. *The relation $\rightarrow_X$ is reflexive and transitive, but not necessarily symmetric.*

It is however possible to derive an equivalence relation from $\rightarrow_X$ by defining the strong connectedness relation $\circlearrowleft_X$ on X by

$$\big(x \circlearrowleft_X y\big) \Leftrightarrow \big((x \rightarrow_X y) \wedge (y \rightarrow_X x)\big) \tag{6}$$

If $x, y \in X$ satisfy $x \circlearrowleft y$, we say that x and y are strongly connected (in X). This notion of strong connectedness is classical the framework of (directed) graphs.

3.3 Semi-connected Components

The notion of strong connectedness leads to equivalence classes of X, namely strongly connected components. The set of these strongly connected components is noted $\mathcal{C}_{\circlearrowleft}[X]$. Similarly, we can define the components that gather elements that are semi-connected.

Definition 10 (Semi-connected components). *Let $x \in X$. The* semi-connected component of X of basepoint x *is the subset of X defined by*

$$\mathcal{C}^x_{\rightarrow}(X) = \{y \in X \mid x \rightarrow_X y\} \tag{7}$$

The set of all the semi-connected components of X is noted $\mathcal{C}_{\rightarrow}[X]$.

By opposition to connected and strongly connected components, the semi-connected components of X do not necessarily form a partition of X.

Property 11. *$\mathcal{C}_{\rightarrow}[X]$ is a cover of X,* i.e., *we have $\emptyset \notin \mathcal{C}_{\rightarrow}[X]$ and $X = \bigcup \mathcal{C}_{\rightarrow}[X]$.*

Indeed, it may happen that distinct semi-connected components have a nonempty intersection.

3.4 Links between Semi-connected and Strongly Connected Components

As stated by Def. 10, a semi-connected component is generated by a specific element, namely its basepoint. This basepoint is not necessarily unique: it may happen that $\mathcal{C}^x_{\rightarrow}(X) = \mathcal{C}^y_{\rightarrow}(X)$ for $x \neq y$. However, it is plain that such basepoints x and y are then strongly connected. From this fact, we straightforwardly derive the following property.

Property 12. *There exists a bijection between $\mathcal{C}_{\circlearrowleft}[X]$ and $\mathcal{C}_{\rightarrow}[X]$, expressed, for all $x, y \in X$, by*

$$\left(C^x_{\rightarrow}(X) = C^y_{\rightarrow}(X)\right) \Leftrightarrow \left(C^x_{\circlearrowleft}(X) = C^y_{\circlearrowleft}(X)\right) \tag{8}$$

Moreover, the partition $\mathcal{C}_{\circlearrowleft}[X]$ refines the cover $\mathcal{C}_{\rightarrow}[X]$.

Property 13. *Let $C \in \mathcal{C}_{\rightarrow}[X]$ be a semi-connected component. There exists a nonempty subset $\mathcal{P} \subseteq \mathcal{C}_{\circlearrowleft}[X]$ of strongly connected components such that $\mathcal{P}$ is a partition of C. In particular, $\mathcal{P}$ is defined as*

$$\mathcal{P} = \{C^x_{\circlearrowleft}[X] \mid x \in C\} \tag{9}$$

From the very definitions of $\rightarrow_X$ and $\circlearrowleft_X$, we finally derive the following property, that describes the structure of $\mathcal{C}_{\rightarrow}[X]$ induced by $\subseteq$, with respect to $\mathcal{C}_{\circlearrowleft}[X]$.

Property 14. *Let $C \in \mathcal{C}_{\rightarrow}[X]$ and $\mathcal{P} \subseteq \mathcal{C}_{\circlearrowleft}[X]$ be defined as above. Let $\mathcal{Q} \subseteq \mathcal{C}_{\rightarrow}[X]$ be the subset of semi-connected components defined by*

$$\mathcal{Q} = \{C^x_{\rightarrow}[X] \mid C^x_{\circlearrowleft}[X] \in \mathcal{P}\} = \{C^x_{\rightarrow}[X] \mid x \in C\} \tag{10}$$

The Hasse diagram of the partially ordered set $(\mathcal{Q}, \subseteq)$ is a directed acyclic graph (DAG), but not a tree in general. The maximum of $(\mathcal{Q}, \subseteq)$ is C, while its minimal elements belong to $\mathcal{P}$, and thus to $\mathcal{C}_{\circlearrowleft}[X]$.

4 Semi-connected Components Hierarchies

In this section, we still suppose that X is equipped with a semi-adjacency $\rightharpoonup_X$, which induces the semi-connectedness relation $\rightarrow_X$ and the strong connectedness relation $\circlearrowleft_X$.

4.1 Properties of Semi-connected Components Hierarchies

Similarly to Sec. 2.2, we discuss here the effects of refining $(X, \rightharpoonup_X)$. Once again, this refinement can be done in two ways: (i) by defining $(Y, \rightharpoonup_Y)$ such that $Y \subseteq X$ and $\rightharpoonup_Y = \rightharpoonup_X \cap (Y \times Y)$ (Figs. 1, 3), or (ii), be defining $(X, \overset{\blacktriangle}{\rightharpoonup}_X)$ such that $\overset{\blacktriangle}{\rightharpoonup}_X \subseteq \rightharpoonup_X$.

Remark 15. *As in Rem. 2, $(Y, \rightharpoonup_Y)$ is a (directed) subgraph of $(X, \rightharpoonup_X)$, while $(X, \overset{\blacktriangle}{\rightharpoonup}_X)$ is a (directed) partial graph of $(X, \rightharpoonup_X)$.*

However, under the current hypotheses, the results of Prop. 3 are no longer totally valid. First, we have the following property that “extends” Eq. (1).

Property 16. *Let $x \in Y$ (resp. $x \in X$). Let $C = C^x_{\rightarrow}(Y) \in \mathcal{C}_{\rightarrow}[Y]$ (resp. $C = C^x_{\overset{\blacktriangle}{\rightarrow}}(X) \in \mathcal{C}_{\overset{\blacktriangle}{\rightarrow}}[X]$), and $C^x_{\rightarrow}(X) \in \mathcal{C}_{\rightarrow}[X]$ be the semi-connected components of basepoint x. We have*

$$C^x_{\rightarrow}(X) = \min_{\subseteq}\{K \in \mathcal{C}_{\rightarrow}[X] \mid C \subseteq K\} \tag{11}$$

Remark 17. *As in Prop. 3, this property guarantees that there exists an inclusion relation between the semi-connected components of same basepoint x between $C_{\rightarrow}[Y]$ (resp. $C_{\overset{\blacktriangle}{\rightarrow}}[X]$) and $C_{\rightarrow}[X]$. However, contrarily to Prop. 3, it does not guarantee that for any given x, $C^x_{\rightarrow}(X)$ is the only semi-connected component that satisfies this inclusion relation. Nevertheless, it states that any other semi-connected component that has the same property also includes $C^x_{\rightarrow}(X)$.*

Still by comparison to Prop. 3, the analogue of Eq. (2) is now no longer satisfied.

Property 18. *With the same hypotheses as above, for any $K \in C_{\rightarrow}[X]$, we have*

$$\big(K \cap C \neq \emptyset\big) \nRightarrow \big(K \supseteq C^x_{\rightarrow}(X)\big) \tag{12}$$

As a corollary, Prop. 4, cannot be generalised to the case of semi-connected components. Indeed, as stated by the following property, *semi-connected components hierarchies are not organised as trees, but as DAGs* (Fig. 4(d)).

Property 19. *Let $(X_i)_{i=0}^k$ (resp. $(\rightharpoonup_i)_{i=0}^k$) ($k \geqslant 0$) be such that $X_0 = X$ (resp. $\rightharpoonup_0 = \rightharpoonup_X$) and $X_{i+1} \subseteq X_i$ (resp. $\rightharpoonup_{i+1} \subseteq \rightharpoonup_i$) for all $i \in [\![0, k-1]\!]$. Let us consider the partially ordered multiset $(\bigcup_{i=0}^k C_{\rightarrow}[X_i], \sqsubseteq)$ (resp. $(\bigcup_{i=0}^k C_{\rightarrow_i}[X], \sqsubseteq)$) (with $\sqsubseteq$ defined as in Eq. (3)). For any $K, K' \in \bigcup_{i=0}^k C_{\rightarrow}[X_i]$ (resp. $\bigcup_{i=0}^k C_{\rightarrow_i}[X]$), we have*

$$\big(K \cap K' \neq \emptyset\big) \nRightarrow \big((K \sqsubseteq K') \vee (K' \sqsubseteq K)\big) \tag{13}$$

Nevertheless, the intersection implies (under certain hypotheses) the inclusion.

Property 20. *Let $x \in Y$ (resp. $x \in X$), and $y \in X$. Let $C = C^x_{\rightarrow}(Y) \in C_{\rightarrow}[Y]$ (resp. $C = C^x_{\overset{\blacktriangle}{\rightarrow}}(X) \in C_{\overset{\blacktriangle}{\rightarrow}}[X]$). Let $C^y_{\rightarrow}(X) \in C_{\rightarrow}[X]$. We have*

$$\big(x \in C \cap C^y_{\rightarrow}(X)\big) \Rightarrow \big(C \subseteq C^y_{\rightarrow}(X)\big) \tag{14}$$

4.2 Properties of Strongly Connected Components Hierarchies

Unlike semi-connected components, strongly connected components are defined as equivalence classes. Thus, they present common intrinsic properties with connected components (Fig. 2). In particular, Props. 3 and 4 can be extended to their case. (Note that $C_{\overset{\blacktriangle}{\circlearrowleft}}[X], C_{\circlearrowleft}[Y]$ are defined the same way as $C_{\overset{\blacktriangle}{\rightleftarrows}}[X], C_{\rightleftarrows}[Y]$ and $C_{\overset{\blacktriangle}{\rightarrow}}[X], C_{\rightarrow}[Y]$.)

Property 21. *Let $x \in Y$ (resp. $x \in X$). Let $C = C^x_{\circlearrowleft}(Y) \in C_{\circlearrowleft}[Y]$ (resp. $C = C^x_{\overset{\blacktriangle}{\circlearrowleft}}(X) \in C_{\overset{\blacktriangle}{\circlearrowleft}}[X]$) be the unique strongly connected component containing x. Then, there exists a unique $C^x_{\circlearrowleft}(X) \in C_{\circlearrowleft}[X]$ that intersects (and actually includes) C.*

Property 22. *Let $(X_i)_{i=0}^k$ (resp. $(\rightharpoonup_i)_{i=0}^k$) ($k \geqslant 0$) be such that $X_0 = X$ (resp. $\rightharpoonup_0 = \rightharpoonup_X$) and $X_{i+1} \subseteq X_i$ (resp. $\rightharpoonup_{i+1} \subseteq \rightharpoonup_i$) for all $i \in [\![0, k-1]\!]$. Let us consider the partially ordered multiset $(\bigcup_{i=0}^k C_{\circlearrowleft}[X_i], \sqsubseteq)$ (resp. $(\bigcup_{i=0}^k C_{\circlearrowleft_i}[X], \sqsubseteq)$) (with $\sqsubseteq$ defined as in Eq. (3)). For any two strongly connected components K, K' of this set, we have*

$$\big(K \cap K' \neq \emptyset\big) \Rightarrow \big((K \sqsubseteq K') \vee (K' \sqsubseteq K)\big) \tag{15}$$

Then, by progressively refining X into successive subsets, or $\rightharpoonup_X$ into successive subrelations, we can build strongly connected components hierarchies as trees (Fig. 4(a)).

4.3 Semi-connected Components Hierarchies as Enriched Strongly Connected Components Hierarchies

On one hand, it has been observed in Sec. 4.1, that the semi-connected components hierarchies induced by progressively refining $(X, \rightharpoonup_X)$, have a structure which cannot be trivially handled. Indeed, these (cover) hierarchies are DAGs, due to inclusions or intersections (without inclusion) between different components at same levels.

On the other hand, it has been observed in Sec. 4.2, that the strongly connected components hierarchies, induced by the very same process, have a much simpler structure. Indeed, these (partition) hierarchies are trees.

Based on the bijection (Prop. 12) that exists between semi-connected and strongly connected components, it is however possible to model the (DAG) hierarchy of semi-connected components as a (tree) hierarchy of the associated strongly connected components, enriched at each level by a "local" DAG that represents the inclusion relation between the semi-connected components of this level. This model is formally expressed by the following proposition.

Proposition 23. *Let $\alpha,\beta \in \llbracket 0,k \rrbracket$, with $\alpha \leqslant \beta$. Let $x,y \in X$. Let $C^x_{\to}(X_\alpha), C^y_{\to}(X_\beta) \in \bigcup_{i=0}^k C_{\to}[X_i]$ (resp. $C^x_{\to_\alpha}(X), C^y_{\to_\beta}(X) \in \bigcup_{i=0}^k C_{\to_i}[X]$). Then, we have*

$$\begin{gathered}\left(C^y_{\circlearrowleft}(X_\beta) \subseteq C^x_{\circlearrowleft}(X_\alpha)\right) \Rightarrow \left(C^y_{\to}(X_\beta) \sqsubseteq C^x_{\to}(X_\alpha)\right) \\ \textit{(resp. } \left(C^y_{\circlearrowleft_\beta}(X) \subseteq C^x_{\circlearrowleft_\alpha}(X)\right) \Rightarrow \left(C^y_{\to_\beta}(X) \sqsubseteq C^x_{\to_\alpha}(X)\right) \textit{)}\end{gathered} \tag{16}$$

and

$$\begin{gathered}\left(C^y_{\to}(X_\beta) \sqsubseteq C^x_{\to}(X_\alpha)\right) \Rightarrow \Big(\left(C^y_{\circlearrowleft}(X_\beta) \subseteq C^y_{\circlearrowleft}(X_\alpha)\right) \wedge \left(C^y_{\to}(X_\alpha) \subseteq C^x_{\to}(X_\alpha)\right)\Big) \\ \textit{(resp. } \left(C^y_{\to_\beta}(X) \sqsubseteq C^x_{\to_\alpha}(X)\right) \Rightarrow \Big(\left(C^y_{\circlearrowleft_\beta}(X) \subseteq C^y_{\circlearrowleft_\alpha}(X)\right) \wedge \left(C^y_{\to_\alpha}(X) \subseteq C^x_{\to_\alpha}(X)\right)\Big) \textit{)}\end{gathered} \tag{17}$$

Proof. Let us suppose that $C^y_{\circlearrowleft}(X_\beta) \subseteq C^x_{\circlearrowleft}(X_\alpha)$ (resp. $C^y_{\circlearrowleft_\beta}(X) \subseteq C^x_{\circlearrowleft_\alpha}(X)$). Then, we have $y \in C^x_{\circlearrowleft}(X_\alpha)$ (resp. $y \in C^x_{\circlearrowleft_\alpha}(X)$), and thus $C^y_{\circlearrowleft}(X_\alpha) = C^x_{\circlearrowleft}(X_\alpha)$ (resp. $C^y_{\circlearrowleft_\alpha}(X) = C^x_{\circlearrowleft_\alpha}(X)$). From Eq. (8) (Prop. 12), we then have $C^y_{\to}(X_\alpha) = C^x_{\to}(X_\alpha)$ (resp. $C^y_{\to_\alpha}(X) = C^x_{\to_\alpha}(X)$). From the definition of X_α and X_β (resp. $\rightharpoonup_\alpha$ and $\rightharpoonup_\beta$) (Prop. 4), and the fact that $\alpha \leqslant \beta$, we straightforwardly have $C^y_{\to}(X_\beta) \subseteq C^y_{\to}(X_\alpha)$ (resp. $C^y_{\to_\beta}(X) \subseteq C^y_{\to_\alpha}(X)$), and then $C^y_{\to}(X_\beta) \subseteq C^x_{\to}(X_\alpha)$ (resp. $C^y_{\to_\beta}(X) \subseteq C^x_{\to_\alpha}(X)$). Eq. (16) then follows from the definition of $\sqsubseteq$, provided in Eq. (3) (Prop. 4).

Let us now suppose that $C^y_{\to}(X_\beta) \sqsubseteq C^x_{\to}(X_\alpha)$ (resp. $C^y_{\to_\beta}(X) \sqsubseteq C^x_{\to_\alpha}(X)$). From the definition of $\sqsubseteq$, we then have $C^y_{\to}(X_\beta) \subseteq C^x_{\to}(X_\alpha)$ (resp. $C^y_{\to_\beta}(X) \subseteq C^x_{\to_\alpha}(X)$). Let us consider $C^y_{\to}(X_\alpha)$ (resp. $C^y_{\to_\alpha}(X)$); this set exists since $y \in C^y_{\to}(X_\beta) \subseteq C^x_{\to}(X_\alpha) \subseteq X_\alpha$ (resp. $y \in C^y_{\to_\beta}(X) \subseteq C^x_{\to_\alpha}(X) \subseteq X$). From the very definition of semi-connectedness (Def. 10), $y \in C^x_{\to}(X_\alpha)$ (resp. $y \in C^x_{\to_\alpha}(X)$) implies that $C^y_{\to}(X_\alpha) \subseteq C^x_{\to}(X_\alpha)$ (resp. $C^y_{\to_\alpha}(X) \subseteq C^x_{\to_\alpha}(X)$), while $X_\beta \subseteq X_\alpha$ (resp. $\rightharpoonup_\beta \subseteq \rightharpoonup_\alpha$) implies that $C^y_{\to}(X_\beta) \subseteq C^y_{\to}(X_\alpha)$ (resp. $C^y_{\to_\beta}(X) \subseteq C^y_{\to_\alpha}(X)$). Finally, we derive from Eq. (8) (Prop. 12) that $C^y_{\circlearrowleft}(X_\beta) \subseteq C^y_{\circlearrowleft}(X_\alpha)$ (resp. $C^y_{\circlearrowleft_\beta}(X) \subseteq C^y_{\circlearrowleft_\alpha}(X)$), and Eq. (17) then follows. ■

Remark 24. *Practically, this proposition proves that the standard relation $\sqsubseteq$, that models the inclusion relation between the semi-connected components at distinct levels of a hierarchy (Fig. 4(d)), can be conveniently handled by simultaneously using the inclusion relation between the strongly connected components associated to the semi-connected ones, and the inclusion relation between the semi-connected components of same level in the hierarchy (Fig. 4(c)). This representation, which is formalised in Diagram* (18), *has the two following virtues:* (i) *it is information lossless, with respect to $\sqsubseteq$; and* (ii) *it replaces the Hasse diagram of $\sqsubseteq$, which is a complex DAG, by a tree structure that is enriched by "local" simple DAGs at each level of the tree. Moreover, its complexity is not excessive by comparison to the Hasse diagram of $\sqsubseteq$.*

$$\begin{tikzcd}
C^x_{\rightarrow}(X_\alpha) & C^y_{\rightarrow}(X_\alpha) \arrow[l, "\subseteq"'] \arrow[r, leftrightarrow, "\sim"] & C^y_{\circlearrowleft}(X_\alpha) & C^x_{\rightarrow_\alpha}(X) & C^y_{\rightarrow_\alpha}(X) \arrow[l, "\subseteq"'] \arrow[r, leftrightarrow, "\sim"] & C^y_{\circlearrowleft_\alpha}(X) \\
C^y_{\rightarrow}(X_\beta) \arrow[u, "\sqsubseteq"'] \arrow[rr, leftrightarrow, "\sim"] & & C^y_{\circlearrowleft}(X_\beta) \arrow[u, "\subseteq"'] & C^y_{\rightarrow_\beta}(X) \arrow[u, "\sqsubseteq"'] \arrow[rr, leftrightarrow, "\sim"] & & C^y_{\circlearrowleft_\beta}(X) \arrow[u, "\subseteq"']
\end{tikzcd} \qquad (18)$$

5 Extending Standard Tree Structures

The "enriched trees" defined above are compliant with standard partition hierarchies when the considered semi-adjacency is indeed an adjacency.

Property 25. *If $\rightharpoonup_X$ is symmetric, we have $C_{\rightarrow}[X] = C_{\circlearrowleft}[X]$ ($= C_{\leftrightarrows}[X]$ by considering $\rightharpoonup_X$ as an adjacency relation). In such conditions, the Hasse diagram induced by $\sqsubseteq$, and the associated enriched tree are equal and both have a tree structure.*

In a reverse way, we show how some standard (total and partial) partition hierarchies can be generalised to handle semi-connected components hierarchies.

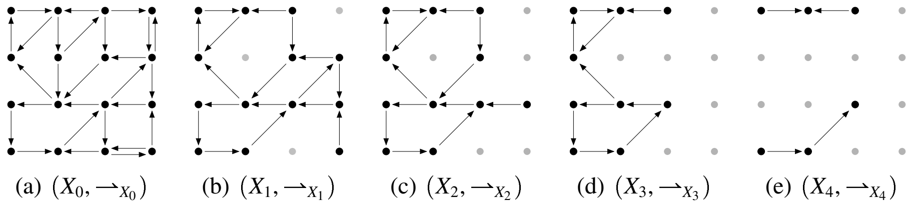
5.1 Partial Partition Hierarchies and Component-Trees

We consider the spatial way to refine $(X, \rightharpoonup_X)$. With this *modus operandi*, the semi-connected components, present in successive subsets, have the same lower elements.

Property 26. *Let $K \in C_{\rightarrow}[X] \cap C_{\rightarrow}[Y]$. Then we have $\{K' \in C_{\rightarrow}[X] \mid K' \subseteq K\} = \{K' \in C_{\rightarrow}[Y] \mid K' \subseteq K\}$.*

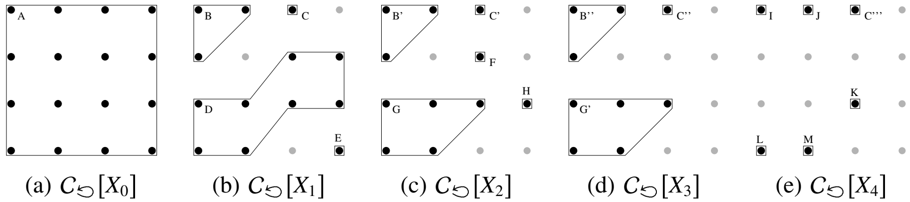
Two examples of such semi-connected components are provided in Fig. 4(c). These (sets of) components, namely B, B' and B" (resp. GB' and G'B"), can be unified into a single semi-connected component B" (resp. G'B"), as it is already trivially done in Fig. 4(d) (where B, B', B" and GB' and G'B" would form two chains, respectively, otherwise).

Remark 27. *Prop. 26 implies that the partially ordered multiset $(\bigcup_{i=0}^{k} C_{\rightarrow}[X_i], \sqsubseteq)$ can be handled as a partially ordered* set. *However, it is not sufficient to claim that there exists an equivalence between the set $\bigcup_{i=0}^{k} C_{\rightarrow}[X_i]$ of all the semi-connected components, and the set $\bigcup_{i=0}^{k} C_{\circlearrowleft}[X_i]$ of all the strongly connected components. In particular, the semi-connected components CDB, C'FGB', C"B" and C"'J of Fig. 4 straightforwardly provide a counter-example to that claim.*



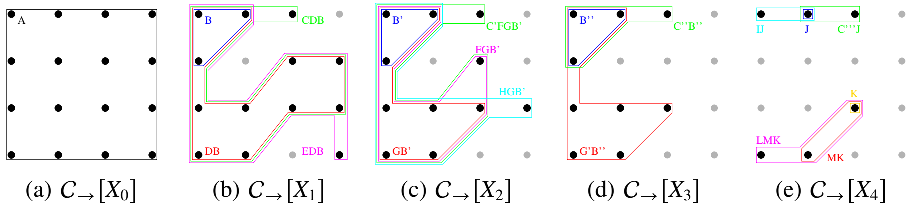
(a) $(X_0, \rightharpoonup_{X_0})$ (b) $(X_1, \rightharpoonup_{X_1})$ (c) $(X_2, \rightharpoonup_{X_2})$ (d) $(X_3, \rightharpoonup_{X_3})$ (e) $(X_4, \rightharpoonup_{X_4})$

Fig. 1. Subsets X_i of a set X, equipped with subrelations $\rightharpoonup_{X_i}$ of a semi-adjacency relation $\rightharpoonup$, for $i = 0$ to 4, with $X_{i+1} \subseteq X_i$ for all $i \in [\![0, 3]\!]$, and $\rightharpoonup_{X_i} = \rightharpoonup \cap (X_i \times X_i)$ for all $i \in [\![0, 4]\!]$



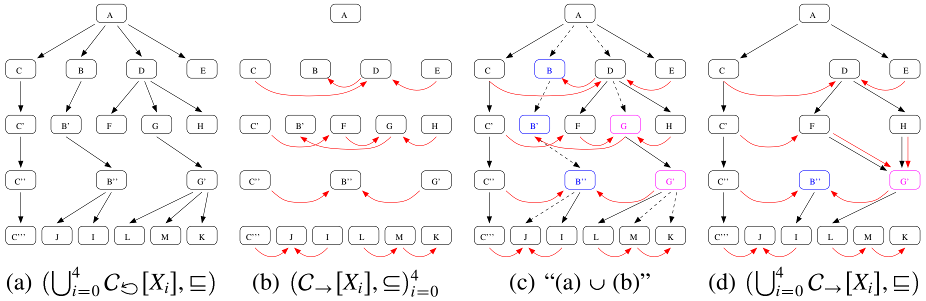
(a) $C_{\circlearrowleft}[X_0]$ (b) $C_{\circlearrowleft}[X_1]$ (c) $C_{\circlearrowleft}[X_2]$ (d) $C_{\circlearrowleft}[X_3]$ (e) $C_{\circlearrowleft}[X_4]$

Fig. 2. The strongly connected components of X_i. Each one is labeled by a capital letter: A, B, C, etc. When a component Z appears in several $C_{\circlearrowleft}[X_i]$, it is successively labeled as Z, Z', Z'', etc.



(a) $C_{\rightarrow}[X_0]$ (b) $C_{\rightarrow}[X_1]$ (c) $C_{\rightarrow}[X_2]$ (d) $C_{\rightarrow}[X_3]$ (e) $C_{\rightarrow}[X_4]$

Fig. 3. The semi-connected components of X_i. Each one is labeled by the capital letters corresponding to the strongly connected components that form its partition: A, CDB, C'FGB', etc.



(a) $(\bigcup_{i=0}^{4} C_{\circlearrowleft}[X_i], \sqsubseteq)$ (b) $(C_{\rightarrow}[X_i], \subseteq)_{i=0}^{4}$ (c) "(a) ∪ (b)" (d) $(\bigcup_{i=0}^{4} C_{\rightarrow}[X_i], \sqsubseteq)$

Fig. 4. (a) Hasse diagram for $\sqsubseteq$ of all the $C_{\circlearrowleft}[X_i]$. (b) Hasse diagrams for $\subseteq$ of each $C_{\rightarrow}[X_i]$. (c) Proposed structure (enriched tree), *i.e.*, the fusion of (a) and (b). (d) Hasse diagram for $\sqsubseteq$ (DAG) of all the $C_{\rightarrow}[X_i]$, modeled by (c). The dashed arrows in (c) are "extra" links with respect to (d). The blue and magenta nodes are elements that may be collapsed in (c) without loss of information. (b–d) For the sake of readability, the nodes are labeled by their first letter, with respect to Fig. 3.

From this remark, we can derive that it is possible to extend the notion of component-tree [19,8], but only up to an equivalence between equal strongly connected components that provide basepoints for semi-connected components.

5.2 Total Partition Hierarchies and Hierarchical Connectivities

We now consider the structural way to refine $(X, \rightharpoonup_X)$.

Remark 28. *By following this* modus operandi*, the semi-connected components that are present in successive subsets do not necessarily have the same lower elements. This difference with Rem. 27 derives from the fact that the semi-connected components of* $C_{\rightharpoonup_i}[X]$ *are composed of elements of* X*, but are actually defined with respect to* $\rightharpoonup_i$.

Practically it is possible to extend (still up to an equivalence) some notions of partition-tree, *e.g.*, those modelling hierarchical connectivities in fuzzy paradigms [17,1].

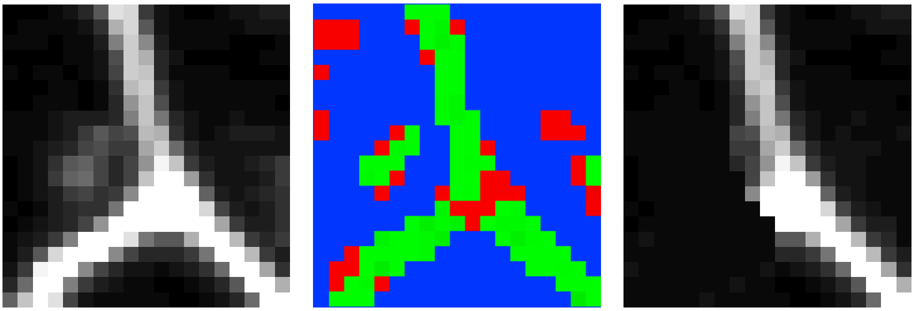
6 Application Example

Basically, the enriched tree, that models a semi-connected components hierarchy, can be built in two steps. The first step consists of computing the hierarchy of the strongly connected components. This can be done by considering an approach based on Tarjan's algorithm. The second step consists of computing, at each level of the tree, the links derived from the inclusion between the semi-connected components. They simply correspond to the remaining semi-adjacency links between the strongly connected components that model the semi-connected ones. A complete algorithmic discussion is beyond the scope of this article, and will be developed in details in further works.

We finally propose a (simple) example whose purpose is to illustrate the relevance of the notion of semi-connectedness, and its methodological usefulness. Let us consider a digital grey-level image $\mathcal{I} : X \to V$ (Fig. 5(a)), that visualises neurites. A Hessian filter can be applied on $\mathcal{I}$, to classify the pixels as linear (L, in green), blobs (B, in red), and others (O, in blue) (Fig. 5(b)). From this classification, the standard 4-adjacency relation defined on $X \subseteq \mathbb{Z}^2$ can be restricted to a semi-adjacency relation $\rightharpoonup_X$ defined by $x \rightharpoonup_X y$ iff x and y belong to the same class, or $x \in O$ or B while $y \in L$.

From the semi-connected components hierarchy of $\mathcal{I}$, with respect to $\rightharpoonup_X$ (which is an extension of a component-tree), we can filter the semi-connected components presenting a linear shape. For instance, in Fig. 5(c), two of the three linear patterns have been preserved, while a third one has been removed due to its orientation.

Of course, this example is of limited purpose, since the crisp classification of X into three classes strongly constrains the space of the possible results. A more satisfactory solution may be to perform a fuzzy classification, that would lead to define semi-adjacencies not only with respect to the level sets of I, but also to the level sets of the fuzzy classification scores. Such perspective works may be relevantly developed in the framework of hypertrees [12]. Beyond these considerations, one may notice that with a standard component-tree, the three linear patterns would have been merged in the same connected component, forbidding (i) linearity characterisation, and (ii) splitting of the two linear patterns of highest intensity. The proposed example, despite its simplicity, then clearly illustrates the potential usefulness of semi-connected filtering.



(a) Image $\mathcal{I} : X \to V$ (b) Classification $\{L, B, O\}$ (c) Filtered image

Fig. 5. Neurite filtering from semi-connected components hierarchy (see text)

7 Conclusion

This work provides first results which demonstrate that cover hierarchies derived from semi-adjacency (i) can be handled via (enriched) tree structures; (ii) provide a way to generalise classical structures such as component-trees and partition-trees; and (iii) may be involved in image processing tasks, in the framework of connected operators.

From a theoretical point of view, the links that may exist between such hierarchies and those induced by hyperconnections [14] could be explored. The relationships with other non-tree hierarchies [13] are also worthy of investigation. More generally, the notion of semi-connection should be axiomatically formalised, beyond the sole framework of graphs, similarly to the proposal of Serra [21] for connections.

From a methodological point of view, the extension of segmentation paradigms based on optimal tree-cuts [6,23] could be considered, with challenges related to algorithmic complexities. New operators could also be designed to provide “disconnection” filters, that may be seen as dual operators with respect to reconnection filters [5].

References

1. Bloch, I., Maître, H., Anvar, M.: Fuzzy adjacency between image objects. Int. J. Uncertain. Fuzz. 5, 615–654 (1997)
2. Braga-Neto, U., Goutsias, J.K.: A theoretical tour of connectivity in image processing and analysis. J. Math. Imaging Vis. 19, 5–31 (2003)
3. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Minimum spanning forests and the drop of water principle. IEEE T. Pattern Anal. 31, 1362–1374 (2009)
4. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Thinnings, shortest path forests, and topological watersheds. IEEE T. Pattern Anal. 32, 925–939 (2010)
5. Dufour, A., Tankyevych, O., Naegel, B., Talbot, H., Ronse, C., Baruthio, J., Dokládal, P., Passat, N.: Filtering and segmentation of 3D angiographic data: Advances based on mathematical morphology. Med. Image Anal. (2012), doi:10.1016/j.media.2012.08.004 (in press)
6. Guigues, L., Cocquerez, J.-P., Le Men, H.: Scale-sets image analysis. Int. J. Comput. Vision 68, 289–317 (2006)
7. Heijmans, H.J.A.M.: Connected morphological operators for binary images. Comput. Vis. Image Und. 73, 99–120 (1999)

8. Jones, R.: Connected filtering and segmentation using component trees. Comput. Vis. Image Und. 75, 215–228 (1999)
9. Meyer, F., Najman, L.: Segmentation, minimum spanning tree and hierarchies. In: Najman, L., Talbot, H. (eds.) Mathematical Morphology: From Theory to Applications, pp. 229–261. ISTE/J. Wiley & Sons (2010)
10. Najman, L.: On the equivalence between hierarchical segmentations and ultrametric watersheds. J. Math. Imaging Vis. 40, 231–247 (2011)
11. Ouzounis, G.K., Wilkinson, M.H.F.: Mask-based second-generation connectivity and attribute filters. IEEE T. Pattern Anal. 29, 990–1004 (2007)
12. Passat, N., Naegel, B.: Component-hypertrees for image segmentation. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 284–295. Springer, Heidelberg (2011)
13. Passat, N., Naegel, B.: Component-trees and multivalued images: Structural properties. Technical report, INRIA-00611714 (2012), url: `http://hal.inria.fr/inria-00611714`
14. Perret, B., Lefèvre, S., Collet, C., Slezak, E.: Hyperconnections and hierarchical representations for grayscale and multiband image processing. IEEE T. Image Proc. 21, 14–27 (2012)
15. Ronse, C.: Set-theoretical algebraic approaches to connectivity in continuous or digital spaces. J. Math. Imaging Vis. 8, 41–58 (1998)
16. Rosenfeld, A.: Connectivity in digital pictures. J. Assoc. Comput. Mach. 17, 146–160 (1970)
17. Rosenfeld, A.: Fuzzy digital topology. Inform. Control 40, 76–87 (1979)
18. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. IEEE T. Image Proc. 9, 561–576 (2000)
19. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE T. Image Proc. 7, 555–570 (1998)
20. Salembier, P., Wilkinson, M.H.F.: Connected operators: A review of region-based morphological image processing techniques. IEEE Signal Proc. Mag. 26, 136–157 (2009)
21. Serra, J.: Mathematical morphology for Boolean lattices. In: Serra, J. (ed.) Image Analysis and Mathematical Morphology, II: Theoretical Advances, pp. 37–58. Academic Press, London (1988)
22. Serra, J.: Connectivity on complete lattices. J. Math. Imaging Vis. 9, 231–251 (1998)
23. Serra, J.: Tutorial on connective morphology. IEEE J. Sel. Top. Signa. 6, 739–752 (2012)
24. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE T. Pattern Anal. 30, 1132–1145 (2008)

Stochastic Morphological Filtering and Bellman-Maslov Chains

Jesús Angulo[1] and Santiago Velasco-Forero[2]

[1] CMM-Centre de Morphologie Mathématique,
Mathématiques et Systèmes, MINES ParisTech, France
[2] ITWM - Fraunhofer Institute, Kaiserlautern, Germany
jesus.angulo@mines-paristech.fr, velascoforero@itwm.fraunhofer.de

Abstract. This paper introduces a probabilistic framework for adaptive morphological dilation and erosion. More precisely our probabilistic formalization is based on using random walk simulations for a stochastic estimation of adaptive and robust morphological operators. Hence, we propose a theoretically sound morphological counterpart of Monte Carlo stochastic filtering. The approach by simulations is inefficient but particularly tailorable for introducing different kinds of adaptability. From a theoretical viewpoint, stochastic morphological operators fit into the framework of Bellman-Maslov chains, the $(\max,+)$-counterpart of Markov chains, which the basis behind the efficient implementations using sparse matrix products.

1 Introduction

Mathematical morphology is a deterministic non-linear methodology for image processing based on a pair of adjoint and dual operators: a dilation (sup-convolution) $(f \oplus b)(\mathbf{x})$ and an erosion (inf-convolution) $(f \ominus b)(\mathbf{x})$ of an image $f : \Omega \to \mathcal{T}$, $\mathbf{x} \mapsto f(\mathbf{x}) = t$, $\Omega \subset \mathbb{Z}^2$ and $\mathcal{T} = \overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$, i.e., $\delta_b(f)(\mathbf{x}) = (f \oplus b)(\mathbf{x}) = \sup_{\mathbf{y} \in \Omega} (f(\mathbf{y}) + b(\mathbf{y} - \mathbf{x}))$ and $\varepsilon_b(f)(\mathbf{x}) = (f \ominus b)(\mathbf{x}) = \inf_{\mathbf{y} \in \Omega} (f(\mathbf{y}) - b(\mathbf{y} + \mathbf{x}))$, where $b(\mathbf{x}) \in \mathcal{F}(\Omega, \mathcal{T})$ is the structuring function which determines the effect of the operator. In the most general setting, the structuring function $b_{\mathbf{z}}(\mathbf{x})$ is variable in space Ω and naturally adapted to each pixel $\mathbf{z}$ w.r.t. the structure and regularity of $f(\mathbf{x})$ in a local neighborhood centered at $\mathbf{z}$. Adaptive mathematical morphology approaches just address the deterministic computation of local structuring function at each pixel.

Aim. In this context, the goal of this work is to introduce a probabilistic framework for adaptive dilation and erosion. More precisely our probabilistic formalization is based on using random walk simulations for a stochastic estimation of adaptive and robust morphological operators. Hence, we propose a theoretically sound morphological counterpart of Monte Carlo stochastic filtering. We briefly discuss also the elements needed for an efficient implementation using $(\max, +)$ linear matrix formulation of Bellman-Maslov chain (counterpart of Markov chain in the $(\max, +)$ algebra. Up to the best of our knowledge, this is the first work

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 171–182, 2013.

which proposes a formulation of stochastic morphological image filtering using random walks. All the results in Section 3 are novel in the state-of-the-art of mathematical morphology.

Related Work. From the theoretical viewpoint of mathematical morphology, our approach fits in the framework of spatially adaptive morphological operators and is connected to other significant previous approaches: i) morphological amoebas [17] and adaptive geodesic neighborhoods [13]; ii) bilateral structuring functions [1]; iii) path openings/closings [7,16]. Our work is related also to algebraic path algorithms in $(\min, +)$ algebra introduced by [12] as a (generalized) Bellman algorithm of shortest paths solved using $(\min, +)$ linear algebra; which have been used for graph-based watershed segmentation [20] and for optimal path image filtering [21].

2 On Stochastic Image Denoising

Random Walks in Images $f \in \mathcal{F}(\Omega, \mathcal{T})$. Random walks are simulated on bounded squared patches $\Omega_{\mathbf{x}}^{\mathrm{Patch}} \subset \Omega$ centered at each pixel $\mathbf{x}$ of the image and of size $(2P+1) \times (2P+1)$ pixels. Each random walk k, starting from the central pixel, i.e., $\mathbf{x}_0 = \mathbf{x}$, is a discrete time ordered sequence of N steps, i.e., $W_{0,N}^k = \{\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_N\}$, visited along the path from $\mathbf{x}_0$ to $\mathbf{x}_N$ where at each step i the walk jumps to another point $i+1$ according to some distribution probability $\Pr(\mathbf{x}_{i+1}|\mathbf{x}_i)$. The number of random walks, or realizations of the random sampling procedure, is M, i.e., the processus is given by $\left\{W_{0,N}^k\right\}_{1 \leq k \leq M}$.

Using a Markovian interpretation, $\Pr(\mathbf{x}_{i+1}|\mathbf{x}_i)$ is viewed as a transition probability of moving from state i to state $i+1$. The transition probabilities in the neighborhood of $\mathbf{x}_i$ depend on distances between the image values $f(\mathbf{x}_i)$. More specifically, let us consider a random walk on the 8−connected discrete grid associated to $\mathbb{Z}^2$: if $\mathbf{x}_i = \mathbf{x}_i^0$ the next point will be one of its neighbors: $\mathbf{x}_{i+1} \in \{\mathbf{x}_i^j\}_{1 \leq j \leq 8}$. Transition probabilities are traditionally defined using Boltzmann-Gibbs distribution depending on a temperature parameter T, which in physics describes the statistical properties of a system in thermodynamic equilibrium. Let us consider the following general model of transition probability

$$\Pr(\mathbf{x}_{i+1}|\mathbf{x}_i) = \frac{1}{Z_T(\mathbf{x}_i)} \exp\left(-\frac{E(\mathbf{x}_{i+1})}{T}\right) \tag{1}$$

where the transition "energy" at pixel $\mathbf{x}_{i+1}$ is given by

$$E(\mathbf{x}_{i+1}) = \lambda\psi(f)(\mathbf{x}_{i+1}) + (1-\lambda)\left[\alpha d_f(\mathbf{x}_i, \mathbf{x}_{i+1})^2 + (1-\alpha) d_f(\mathbf{x}_0, \mathbf{x}_{i+1})^2\right] \tag{2}$$

The temperature T is the scaling parameter; and $Z_T(\mathbf{x}_i)$ is a normalization constant called the partition function, ensuring that the probabilities sum up to one, i.e.,

$$Z_T(\mathbf{x}_i) = \sum_{1 \leq j \leq 8} \exp\left(-\frac{E(\mathbf{x}_i^j)}{T}\right) \tag{3}$$

The partition function encodes how the probabilities are partitioned among the different pixels of the neighborhood, based on their individual energies. It counts the (weighted) number of states a system can occupy: hence if all states are equally probable (equal energies) the partition function is the total number of possible states. We notice that the probability depends inversely on the transition energies, and these energies depend on the distance between two gray-level image pixels (or dissimilarity measure in f) given by $d_f(\mathbf{x}_a, \mathbf{x}_b) = |f(\mathbf{x}_a) - f(\mathbf{x}_b)|$.

Linear (convex) combination parameters $\alpha \in [0,1]$ and $\lambda \in [0,1]$ allows a fine-tuning of the transition energy, which can be separated into three terms:

- $E_1(\mathbf{x}_{i+1}) = \psi(f)(\mathbf{x}_{i+1}) = -f(\mathbf{x}_{i+1})^2$: This term only depends on the corresponding image value and it is higher for low intensity values;
- $E_2(\mathbf{x}_{i+1}) = d_f(\mathbf{x}_i, \mathbf{x}_{i+1})^2$: This is a gradient term which imposes a preference for smooth transitions and avoiding crossing over image edges;
- $E_3(\mathbf{x}_{i+1}) = d_f(\mathbf{x}_0, \mathbf{x}_{i+1})^2$: This term measures the similarity between the origin $\mathbf{x}_0$ and prevents strong changes due to gradual cumulated transitions.

The chain $W_{0,i+1}$ is generated from $W_{0,i}$ by sampling the neighborhood of $\mathbf{x}_i$ and choosing a neighbor with probability $\Pr(\mathbf{x}_{i+1}|\mathbf{x}_i)$. Under the first-order Markov assumption, given the initial probability distribution of source point $\Pr(\mathbf{x}_0)$, the probability of the Markov chain $W_{1,l}$ is given by $\Pr(W_{1,l}|\mathbf{x}_0) = \Pr(\mathbf{x}_0)\prod_{t=1}^{l}\Pr(\mathbf{x}_t|\mathbf{x}_{t-1})$. In our case, all the random walks are initiated at the origin $\mathbf{x}_0$ and consequently $\Pr(\mathbf{x}_0) = 1$. Hence, the probability of the walk on the image at a given step l ($1 \leq l \leq N$) is given by

$$\Pr(W_{1,l}|\mathbf{x}_0) = \prod_{i=1}^{l} \Pr(\mathbf{x}_i|\mathbf{x}_{i-1}) = \prod_{i=1}^{l} \frac{1}{Z_T(\mathbf{x}_{i-1})} \exp\left(-\frac{E(\mathbf{x}_i)}{T}\right) \tag{4}$$

According to that, probabilities in the walk always decrease along the space path, and consequently a term of spatial distance, i.e., $\|\mathbf{x}_i - \mathbf{x}_{i+1}\|$, is not needed to account this effect.

Hence, for a given random walk k, the estimated probability at pixel $\mathbf{x}_i$ is given by $p_k(\mathbf{x}_i) = \Pr\left(W_{1,i}^k|\mathbf{x}_0\right)$. By integrating the M walks, we obtain a Monte Carlo estimation in the x_0-centered patch $\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}$ of the probability density function $pdf_{\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}}(\mathbf{x})$ associated to a source point $\mathbf{x}_0$:

$$pdf_{\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}}(\mathbf{x}_i) = \frac{1}{M}\sum_{k=1}^{M} p_k(\mathbf{x}_i) \tag{5}$$

The pdf can be interpreted as a stochastic kernel used for locally and adaptively filtering the pixel $\mathbf{x}_0$.

Stochastic Image Filtering Using Random Walks. The rationale behind the approach is based on the fact that random walks over small image neighbourhoods using image-adapted transition probabilities provide a good estimation of the appearance of the noise-free pixels. It has been proven [4,11] that it is a stable estimation with a relatively small number of samples M (walks). The idea is denoising pixel $\mathbf{x}_0$ as a probabilistic estimation based on repeating sampling walks $W_{0,i}^k$ starting from the specific pixel $\mathbf{x}_0$, and combining the visited pixels along each walk with weights given by its probabilities $p_k(\mathbf{x}_0)$. More precisely, given the vector of parameters $\Theta = \{P, T, \alpha, \lambda, M, N\}$, the stochastic filter SF is defined as

$$\mathrm{SF}_\Theta(f)(\mathbf{x}_0) = \frac{1}{M}\sum_{k=1}^{M}\left(\frac{\sum_{i=1}^{N} p_k(\mathbf{x}_i)f(\mathbf{x}_i)}{\sum_{i=1}^{N} p_k(\mathbf{x}_i)}\right) = \frac{1}{M}\sum_{k=1}^{M}\sum_{i=1}^{N}\widetilde{p}_k(\mathbf{x}_i)f(\mathbf{x}_i) \qquad (6)$$

where $\widetilde{p}_k(\mathbf{x}_i) = p_k(\mathbf{x}_i)/\sum_{i=1}^{N} p_k(\mathbf{x}_i)$. In addition, using the (normalized) probability density function of the patch centered at $\mathbf{x}_0$ as a stochastic convolution kernel, we have the following equivalent expression:

$$\mathrm{SF}_\Theta(f)(\mathbf{x}_0) = \sum_{\mathbf{y}\in\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}} \widetilde{pdf}_{\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}}(\mathbf{y})f(\mathbf{x}_0 - \mathbf{y}) \qquad (7)$$

where $\widetilde{pdf}_{\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}}(\mathbf{x}_i) = 1/M\sum_{k=1}^{M}\widetilde{p}_k(\mathbf{x}_i)$. The parameters of the stochastic filtering are as follows: P is the half-size of the patch, determining the scale of the filter; N is the length of the walks (we can naturally fix $N = P$ since the longest possible shortest path has a length equal to the patch half size); M number of realizations (as long as the walk is long enough, few realizations provide a reliable sampling), we typically fix $M = 50$; T is the temperature, being the critical parameter which determines the local adaptivity scale of the filter, a good trade-off has been obtained for $T = 0.001$. With respect to the energy parameters, we typically fix for general purpose applications $\alpha = 0.5$; the case $\lambda \neq 0$ is considered below. It is well known that to have a robust estimation at pixel $\mathbf{x}_0$ of a noisy image, the distance between pixels should be computed in a regularized version of the image, i.e., $d_f(\mathbf{x}_a, \mathbf{x}_b) = |\phi(f)(\mathbf{x}_a) - \phi(f)(\mathbf{x}_b)|^2$, where $\phi(f)$ is a (fast) spatially-invariant filter (e.g., a Gaussian or median filter in a small window, reducing the dependence on the noise estimation).

3 Stochastic Dilation and Erosion

3.1 Morphologization Trick

Let us start by introducing a straightforward method to define the counterpart of the convolution in the $(+, \times)$ algebra to the convolution of the $(\max, +)$ and $(\min, +)$ algebras. Based on previous results of the state-of-the-art [19,15,8,1], which justifies a logarithmic connection between linear convolution and sup/inf-convolution, the morphologization trick can be stated as follows:

$$(f * k)(\mathbf{x}) = \int_\Omega f(\mathbf{y})k(\mathbf{y}-\mathbf{x})d\mathbf{y} \longrightarrow \left[\begin{array}{l} b(\mathbf{x}) = \log\left(k(\mathbf{x})\right) \\ \left\{\begin{array}{l} (f \oplus b)(\mathbf{x}) = \sup_{\mathbf{y}\in\Omega}\left[f(\mathbf{y}) + b(\mathbf{y}-\mathbf{x})\right] \\ (f \ominus b)(\mathbf{x}) = \inf_{\mathbf{y}\in\Omega}\left[f(\mathbf{y}) - b(\mathbf{y}+\mathbf{x})\right] \end{array}\right. \end{array}\right.$$

Hence, given a linear convolution with a non-negative kernel $k(\mathbf{x})$, the associated sup-convolution and inf-convolution involves the computation of the logarithm of the kernel in order to obtain the appropriate structuring function $b(\mathbf{x})$. Furthermore, we have the following kernel vs. structuring function related properties: $0 \leq k(\mathbf{x}) \leq 1$, $\forall \mathbf{x} \in \Omega$; $k(\mathbf{0}) = 1$; $\sum_{\mathbf{x}\in\Omega} k(\mathbf{x}) = 1 \iff -\infty \leq b(\mathbf{x}) \leq 0$, $\forall \mathbf{x} \in \Omega$; $b(\mathbf{0}) = 0$; $\sup_{\mathbf{x}\in\Omega} b(\mathbf{x}) = 0$.

3.2 Stochastic Sup/Inf-Convolution Operators

Using the morphologization trick, *stochastic dilation* and *stochastic erosion* of image $f \in \mathcal{F}(\Omega, \overline{\mathbb{R}})$, with parameters $\Theta = \{P, T, \alpha, \lambda, M, N\}$, are respectively given by

$$\mathrm{S}\delta_{\Theta}(f)(\mathbf{x}_0) = \sup_{1\leq k\leq M} \left[\sup_{1\leq i\leq N} \left[f(\mathbf{x}_i) + \mathfrak{p}_k(\mathbf{x}_i) \right] \right] \tag{8}$$

and

$$\mathrm{S}\varepsilon_{\Theta}(f)(\mathbf{x}_0) = \inf_{1\leq k\leq M} \left[\inf_{1\leq i\leq N} \left[f(\mathbf{x}_i) - \mathfrak{p}_k(\mathbf{x}_i) \right] \right] \tag{9}$$

where, using (4), the *stochastic morphological penalization* from walk k is

$$\mathfrak{p}_k(\mathbf{x}_i) = \log\left(p_k(\mathbf{x}_i)\right) = \log\left(\Pr\left(W_{1,i}^k|\mathbf{x}_0\right)\right) = \sum_{m=1}^{i} \log\left(\Pr(\mathbf{x}_m|\mathbf{x}_{m-1})_k\right) \tag{10}$$

Thus, the value of the penalization is the sum of the previous values of the logarithm of the transition probabilities. Let us introduce the notion of *penalization density function* of the patch $\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}$ centered at $\mathbf{x}_0$ as the following function

$$\mathfrak{pdf}_{\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}}(\mathbf{x}) = \begin{cases} \sup_{1\leq k\leq M} \left[\mathfrak{p}_k(\mathbf{x})\right] & \text{if } \mathbf{x} \in \{\text{at least one of the } M \text{ walks}\} \\ -\infty & \text{otherwise} \end{cases} \tag{11}$$

In fact, the penalization density function $\mathfrak{pdf}_{\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}}(\mathbf{x})$ can be considered as the *stochastic structuring function* locally associated to pixel $\mathbf{x}_0$, which allow us to rewrite the stochastic dilation and erosion as spatially variant operators:

$$\mathrm{S}\delta_{\Theta}(f)(\mathbf{x}_0) = \sup_{\mathbf{y}\in\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}} \left[f(\mathbf{x}_0 - \mathbf{y}) + \mathfrak{pdf}_{\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}}(\mathbf{y}) \right] \tag{12}$$

$$\mathrm{S}\varepsilon_{\Theta}(f)(\mathbf{x}_0) = \inf_{\mathbf{y}\in\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}} \left[f(\mathbf{x}_0 - \mathbf{y}) - \mathfrak{pdf}_{\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}}(\mathbf{y}) \right] \tag{13}$$

We just name the logarithm of the transition probabilities as the transition penalization $\mathrm{Pe}(\mathbf{x}_m|\mathbf{x}_{m-1})_k$ and we can now write $\mathfrak{p}_k(\mathbf{x}_i) = \sum_{m=1}^{i} \mathrm{Pe}(\mathbf{x}_m|\mathbf{x}_{m-1})_k$. More precisely, the *transition penalization* (a Maslov measure) between pixels $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ is defined as

$$\mathrm{Pe}(\mathbf{x}_{i+1}|\mathbf{x}_i) = \log\left(\Pr(\mathbf{x}_{i+1}|\mathbf{x}_i)\right) = -\frac{E(\mathbf{x}_{i+1})}{T} + \frac{-T\log\left(\sum_{1\leq j\leq 8}\exp\left(-\frac{E(\mathbf{x}_i^j)}{T}\right)\right)}{T} \tag{14}$$

Interpretation of the Transition Penalization. Let us give a deeper insight of the transition penalization. Considering the classical Maslov quantization results [18], i.e., $\inf(a,b) = \lim_{h\to 0^+} -h\log\left(\exp\left(-a/h\right) + \exp\left(-b/h\right)\right)$, we have, from (14), that for a low temperature, i.e., $T \to 0^+$, the transition penalization is approximately equivalent to

$$\mathrm{Pe}(\mathbf{x}_{i+1}|\mathbf{x}_i) \approx -\left(\frac{E(\mathbf{x}_{i+1})}{T} - \inf_{1\leq j\leq 8}\left[\frac{E(\mathbf{x}_i^j)}{T}\right]\right) = -\left(\frac{E(\mathbf{x}_{i+1})}{T} - \mathfrak{Z}_T(\mathbf{x}_i)\right)$$

where we have introduced the *infimal partition function*

$$\mathfrak{Z}_T(\mathbf{x}_i) = \inf_{1\leq j\leq 8}\left[\frac{E(\mathbf{x}_i^j)}{T}\right] \tag{15}$$

As $\mathbf{x}_{i+1} \in \{\mathbf{x}_i^j\}$, $1 \leq j \leq 8$, we can interpret the infimal partition function $\mathfrak{Z}_T(\mathbf{x}_i)$ (i.e., infimum of energies/temperature) as the morphological counterpart of partition function $Z_T(\mathbf{x}_i)$ (i.e., sum of Boltzman factors) in the Gibbs distribution; in order words, the normalization by subtraction of the minimum value is the morphological counterpart of the normalization by division of the sum. If we remember the expression of the transition "energy", Eq. (2), and taking into account that for images valued in $\mathcal{T} = [0,1]$, we have $-1 \leq E(\mathbf{x}_{i+1}) \leq 1$. Hence, this morphological normalization procedure guarantees that $\left(E(\mathbf{x}_{i+1}) - \inf_{1\leq j\leq 8}\left[E(\mathbf{x}_i^j)\right]\right) \geq 0$ and therefore $-\infty \leq \mathrm{Pe}(\mathbf{x}_{i+1}|\mathbf{x}_i) \leq 0$. This is needed for a Maslov idempotent measure [9].

Coming back to the expression of the stochastic morphological weight for a walk k, it can be rewritten for a low temperature as

$$\mathfrak{p}_k(\mathbf{x}_i)\mid_{T\to 0^+} \approx -\sum_{m=1}^{i}\left(\frac{E(\mathbf{x}_m)}{T} - \mathfrak{Z}_T(\mathbf{x}_{m-1})\right)_k \tag{16}$$

which should be compared with the probability

$$p_k(\mathbf{x}_i) = \prod_{m=1}^{i}\frac{1}{Z_T(\mathbf{x}_{m-1})}\exp\left(-\frac{E(\mathbf{x}_m)}{T}\right)_k. \tag{17}$$

Examples of Stochastic Dilation and Erosion. Fig. 1 reveals the effect of geodesic distance vs. distance-to-origin terms in the transition penalization, which is regulated by the parameter α. In particular, the pure geodesic distance case ($\alpha = 1$) involves dilation and erosion by geodesic balls which produces a strong morphological transformation of the image into a piecewise constant function. When the distance-to-origin is considered exclusively ($\alpha = 0$), the contours and level sets of the image are better preserved and the corresponding stochastic dilation and erosion are more regular.

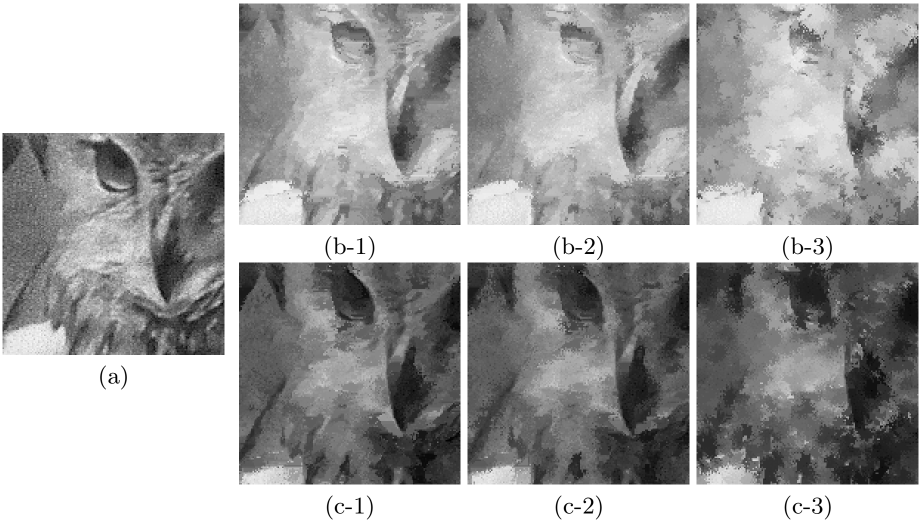


Fig. 1. Stochastic dilation $\mathrm{S}\delta_{\Theta}(f)$ and erosion $\mathrm{S}\varepsilon_{\Theta}(f)$, with $\Theta = \{P, T, \alpha, \lambda, M, N\}$, of original image f given in (a), with $P = N = 21$; $T = 0.001$; $\lambda = 0$ and $M = 50$. First column, $\alpha = 0$; second column, $\alpha = 0.5$; third column, $\alpha = 1$. Row (b-), stochastic dilation; row (c-), stochastic erosion.

Maslovian Random Walk. Stochastic dilation/erosion can be considered as an optimized morphological process based on Monte Carlo simulation of Maslovian random walks. More precisely, we define a Maslovian random walk of length N as a discrete time indexed ordered sequence of points $\mathbf{z}_t$, starting from the point $\mathbf{z}_0$: $\mathfrak{W}_{0,N} = \{\mathbf{z}_0, \mathbf{z}_1, \cdots \mathbf{z}_N\}$ visited along the path from $\mathbf{z}_0$ to $\mathbf{z}_N$ where at each step t the walk jumps to another point $t+1$ to some distribution $\mathrm{Pe}(\mathbf{z}_{t+1}|\mathbf{z}_t)$ viewed as a transition penalization of moving from state t to state $t+1$.

Then, given the initial penalization distribution of initial point $\mathrm{Pe}(\mathbf{z}_0)$, the probability of the Maslovian chain $\mathfrak{W}_{1,l}$ is given by

$$\mathrm{Pe}\left(\mathfrak{W}_{1,l}|\mathbf{z}_0\right) = \mathrm{Pe}(\mathbf{z}_0) + \sum_{t=1}^{l} \mathrm{Pe}(\mathbf{z}_t|\mathbf{z}_{t-1})$$

From a more theoretical viewpoint, our Maslovian random walks are a special type of Bellman-Maslov chains. Bellman-Maslov chains are the counterpart of Markov chains in the (max, +) algebra, and they are mainly used in optimization theory [10,2]. Its theoretical roots are founded on the Maslov idempotent measure theory [18] and decision process theory [6]. It was fully developed in [9] and [3] by the construction of a decision theory at the same level of generality as probability and stochastic processes theory.

The key idea consist in replacing, in the structural axioms of probability theory, the role of the classical semiring $\mathrm{S}_{(+,\times)} = (\mathbb{R}_+, +, \times, 0, 1, \leq)$ of positive real numbers by the idempotent semiring $\mathrm{S}_{(\max,+)} = (\mathbb{R}_- \cup -\infty, \max, +, -\infty, 0, \leq)$.

The dynamics of a Markov chain is linear in the semiring $\mathrm{S}_{(+,\times)}$ and the Bellman-Maslov chain is linear in the idempotent semiring $\mathrm{S}_{(\max,+)}$. This point will be discussed in the Section on efficient implementation.

3.3 Stochastic Adjoint Opening and Closing

The corresponding adjoint stochastic opening to the pair $(\mathrm{S}\delta, \mathrm{S}\varepsilon)$ may be formulated geometrically as a spatially-variant opening using the notion of pulse function [14] of level t at point $\mathbf{p}$ given by

$$i_{(\mathbf{p},t)}(\mathbf{x}) = \begin{cases} t & \mathbf{x} = \mathbf{p} \\ -\infty & \mathbf{x} \neq \mathbf{p} \end{cases}, \ \forall \mathbf{x} \in \Omega, \tag{18}$$

such that any image $g \in \mathcal{F}(\Omega, \mathcal{T})$ can be decomposed into the supremum of its pulses, i.e., $g = \vee\{i_{(\mathbf{x},g(\mathbf{x}))}, \mathbf{x} \in \Omega\}$. Let us introduce also the translate of the function $g(\mathbf{x})$ by the pair $(\mathbf{p}, t)$ as the function $g_{(\mathbf{p},t)}$ whose graph is obtained by translating $\{\mathbf{x}, g(\mathbf{x})\}$ by $\mathbf{p}$ in the first coordinate and by t in the second, i.e., $g_{(\mathbf{p},t)}(\mathbf{y}) = g(\mathbf{y} - \mathbf{p}) + t$. Using the impulse function, the *stochastic opening* $\mathrm{S}\gamma_\Theta(f)$ by adjunction is formulated as

$$\mathrm{S}\gamma_\Theta(f) = \bigvee \left\{ \mathfrak{pdf}_{(\mathbf{p},t);\ \Omega_{\mathbf{p}}^{\mathrm{Patch}}} :\ (\mathbf{p}, t) \in \Omega \times \mathcal{T};\ \mathfrak{pdf}_{(\mathbf{p},t);\ \Omega_{\mathbf{p}}^{\mathrm{Patch}}} \leq f \right\} \tag{19}$$

which is consequently formulated exclusively by translation of the local penalization density functions $\mathfrak{pdf}_{\Omega_{\mathbf{p}}^{\mathrm{Patch}}}(\mathbf{x})$ at each pair $(\mathbf{p}, t)$ and which leads to a straightforward algorithmic implementation. Considering f as a surface embedded in a 3D space, we have the following the geometric interpretation: the subgraph of the opening $\mathrm{S}\gamma_\Theta(f)$ is generated by the zone swept by all "hatted cylinders" $(\mathfrak{pdf}_{\Omega_{\mathbf{p}}^{\mathrm{Patch}}}(\mathbf{y} - \mathbf{p}) + t)$ smaller than f. The *stochastic closing* $\mathrm{S}\varphi_\Theta(f)$ is defined by duality of the opening. Obviously, by construction, stochastic opening and closing are idempotent operators for a given set of local penalization density functions $\{\mathfrak{pdf}_{\Omega_{\mathbf{x}_0}^{\mathrm{Patch}}}(\mathbf{y});\ \forall \mathbf{x}_0 \in \Omega\}$. Examples of stochastic openings and closings are given in Fig. 2, which illustrates the effect of temperature as well as the effect of parameter α on these morphological filters.

3.4 Stochastic Operators as Adaptive Spatially-Variant Ones

Stochastic dilation/erosion are a particular case of adaptive spatially-variant dilation/erosion and consequently, they have the corresponding theoretical properties. Moreover, we can point out the asymptotic connections with other well-known spatially adaptive morphological operators.

Let us start by the morphological amoebas [17]. We remember that the flat amoeba of size r at point x is defined by $A_r(\mathbf{x}) = \{\mathbf{y} \in \Omega :\ \min_{\{\mathbf{x}_1 = \mathbf{x}, \mathbf{x}_2, \cdots, \mathbf{x}_N = \mathbf{y}\}} d_{Amoeba}(\mathbf{x}, \mathbf{y}) < r\}$, where $d_{Amoeba}(\mathbf{x}, \mathbf{y}) = N + \lambda_{\mathrm{Amoeba}} \sum_{i=1}^{N} |f(\mathbf{x}_i) - f(\mathbf{x}_{i+1})|$. Thus, it corresponds to the case of transition energy E_2 (i.e., $\alpha = 1$, $\lambda = 0$): for $M \to +\infty$, with $T = 1/\lambda_{\mathrm{Amoeba}}$. The adapted structuring element

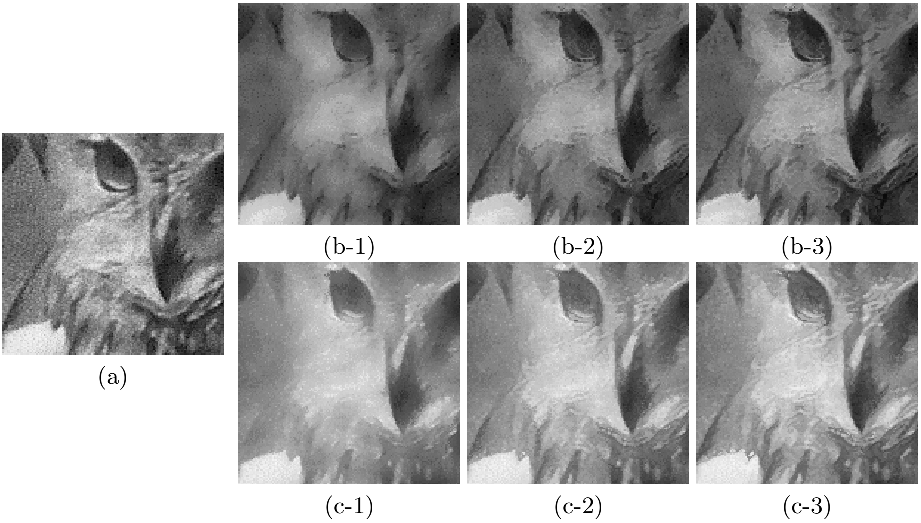


Fig. 2. Stochastic opening $\mathrm{S}\gamma_{\Theta}(f)$ and closing $\mathrm{S}\varphi_{\Theta}(f)$, with $\Theta = \{P, T, \alpha, \lambda, M, N\}$, of original image f given in (a), with $P = N = 21$; $\lambda = 0$; and $M = 50$. First column, $T = 0.01$, $\alpha = 0.5$; second column, $T = 0.001$, $\alpha = 0.5$; third column, $T = 0.001$, $\alpha = 0$. Row (b-), stochastic opening; row (c-), stochastic closing.

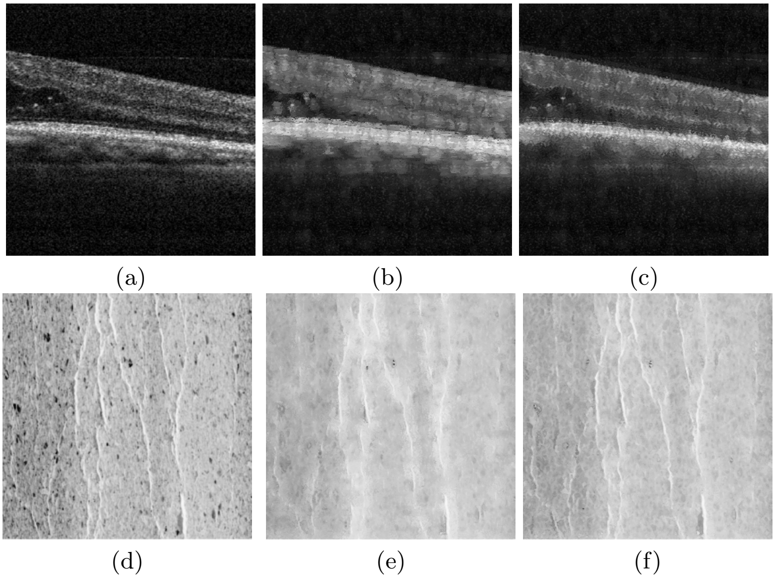


Fig. 3. Top, stochastic dilation $\mathrm{S}\delta_{\Theta}(f)$ in (b) and closing $\mathrm{S}\varphi_{\Theta}(f)$ in (c) of original image f given in (a), with $P = N = 21$; $\alpha = 0$, $\lambda = 0$; and $M = 50$. Bottom, stochastic closing $\mathrm{S}\varphi_{\Theta}(f)$ of original image f given in (d), with $P = N = 21$; $T = 0.001$; and $M = 50$; (e) $\alpha = 0.5$, $\lambda = 0.5$; (f) $\alpha = 0$, $\lambda = 0.5$.

$A_r(\mathbf{x})$ can be obtained by thresholding the penalization distribution function $\mathfrak{pdf}_{\Omega_{\mathbf{x}}^{\text{Patch}}}(\mathbf{z})$ at a given value related to r. As we have discussed above in the paper, the corresponding dilation/erosion produces piecewise constant images. In the case of bilateral morphology [1], the local structuring function, i.e., $b_{\sigma_s,\sigma_i}(\mathbf{x}) = -\|\mathbf{x}-\mathbf{y}\|^2/2\sigma_s^2 - |f(\mathbf{x})-f(\mathbf{y})|^2/2\sigma_i^2$, is related to the transition energy E_3 (i.e., $\alpha = 0$, $\lambda = 0$) with a temperature $T = \sigma_i^2/\sigma_s^2$, but of course including a vertical normalization. The interest in the underlaying stochastic operators for achieving adaptive filtering is clear. The example given in Fig. 3-Top shows how the stochastic dilation and closing are used to morphologically regularize the image, without introducing any blur or distortion of main structures. By morphology regularization, we mean that the structures are asymmetrically simplified, in this case the dark areas are filled in, without modifying the bright ones.

Finally, let consider the example of Fig. 3-Bottom: By including a significant contribution of the energy E_1 (i.e., $\lambda > 0.5$), the corresponding stochastic openings/closing are associated to random walks which correspond roughly to the paths used for the path openings/closings [7,16], which are morphological connected filters that use families of structuring elements consisting of flexible constrained paths which are adapted to local image structures, for which there exists algorithms as efficient as those using the usual families of straight line segments.

4 Towards Efficient Implementation Using Matrix Products

From Markov Chains to Bellman-Maslov Chains. Using results from [9,10], we can state the basics on Bellman-Maslov chains as a parallel of classical Markov chains. Let us construct an $N \times N$ matrix $\Xi(t)$, the transition cost matrix, whose (i,j)-th entry is $\mathfrak{c}_{i,j}(t)$, with $i,j \in \mathcal{I}$, where $\mathfrak{c}_{i,j}(t)$ is the cost of going from state i to j at the t-th time step: $\mathfrak{c}_{i,j}(t) = \mathbb{K}_{t|t-1}(X_t = j|X_{t-1} = i)$, where $\mathbb{K}_{t|t-1}(X_t|X_{t-1})$ is the conditional cost. We denote the cost that the Bellman-Maslov chain reaches the state i at time step t by $\mathfrak{c}_i(t) = \mathbb{K}(X_t = i)$. We can now use a vector $\mathfrak{P}(t) = [\mathfrak{c}_1(t), \cdots, \mathfrak{c}_N(t)]$ to represent the cost distribution of the Bellman-Maslov chain over all states at time t. Note that $\sup_{i\in\mathcal{I}} \mathfrak{c}_i(t) = 0$. The transition cost matrices together with the initial cost distribution completely determine the Bellman-Maslov chain.

Let the initial distribution vector be denoted by $\mathfrak{P}(0)$. Then the distribution of the Bellman-Maslov chain is $\mathfrak{P}(1) = \mathfrak{P}(0) \boxtimes \Xi(1)$, after one step, and is $\mathfrak{P}(t) = \mathfrak{P}(t-1) \boxtimes \Xi(t) = \mathfrak{P}(0) \boxtimes \Xi^t$, where $\Xi^t = \Xi(1) \boxtimes \Xi(2) \boxtimes \cdots \boxtimes \Xi(t)$ is called the t-step transition cost matrix, obtained by $(+,\times)$ product of Bellman-Maslov matrices. More precisely, the product of two matrices A and B in the semimodule $\mathrm{S}_{(\max,+)}$ is a matrix in $\mathcal{M}_{n,l}(\mathrm{S}_{(\max,+)})$ defined by $A \boxtimes B = [\max_{1\leq k\leq p}(C_{i,k} + D_{k,j})]_{i=1,\cdots n;j=1,\cdots l}$. We write also $\underbrace{A \boxtimes A \boxtimes \ldots \boxtimes A}_{n \text{ times}} = A^{\boxtimes n}$. We notice that if the Bellman-Maslov chain is stationary, i.e., $\Xi(1) = \Xi(2) = \cdots = \Xi(t)$, t-step transition cost matrix is the $(\max,+)$-power to t of the initial matrix $\Xi^t = (\Xi(1))^{\boxtimes t}$.

Implementation Using Sparse Algebra. A Maslovian random walk can be viewed as a special type of stationary Bellman-Maslov chain, with $\mathfrak{P}(0) = [0, \cdots, 0]$. Now, we only need to compute the matrix $\Xi = \Xi(1) = \mathrm{Pe}(\mathbf{x}_j|\mathbf{x}_i)$ and its $(\max, +)$ power t to obtain the cost distribution of pixel i as the i-th row of Ξ^t. There exist very efficient numerical algorithms for representing sparse stochastic matrices and to compute their products. Unfortunately, Ξ^t is not a sparse matrix even for small t. Therefore, to use all the power of numerical methods for sparse matrices, we propose to work on the Markov matrix framework.

The algorithm is based on computing the transition probability matrix, and then, to obtain the stochastic structuring functions by the "morphologization trick" of the corresponding column of the t-step matrix. Additionally, we limited ourselves to symmetric stochastic structuring function to fit the framework of "structuring element system" introduced in [22].

5 Conclusions and Perspectives

Stochastic dilation and erosion are naturally formalized by Monte Carlo simulation of Maslovian random walks. The approach by simulation is inefficient but can be particularly tailored to introduce different kinds of adaptability. Efficient implementations, using an optimized product of matrices and parallel computing, lead to relatively fast algorithms. Processing results obtained from stochastic openings/closings are very promising in comparison with other deterministic locally adaptive approach. From a theoretical viewpoint, stochastic morphological operators fit into the framework of Bellman-Maslov chains, the $(\max, +)$-counterpart of Markov chain.

This work can be continued in various ways. We can consider for instance stochastic morphological operators using other "energies", such as distances between structure tensors, distance between covariance matrices, non-local distances, etc. We can also explore the effect of gradual temperature decreasing in iterative operators (morphological counterpart of simulated annealing).

References

1. Angulo, J.: Morphological Bilateral Filtering and Spatially-Variant Adaptive Structuring Functions. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 212–223. Springer, Heidelberg (2011)
2. Akian, M., Quadrat, J.P., Viot, M.: Duality between Probability and Optimization. In: Gunawardena (ed.) Idempotency, Cambridge University Press (1998)
3. Akian, M.: Densities of idempotent measures and large deviations. Trans. of the American Mathematical Society 351(11), 4515–4543 (1999)
4. Azzabou, N., Paragios, N., Guichard, F.: Random walks, constrained multiple hypothesis testing and image enhancement. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 379–390. Springer, Heidelberg (2006)
5. Black, M., Sapiro, G., Marimont, D., Heeger, D.: Robust anisotropic diffusion. IEEE Trans. on Image Processing 7, 421–432 (1998)

6. Bellman, R., Dreyfus, S.: Dynamic Programming and Applications. Dunod, Paris (1965)
7. Buckley, M., Talbot, H.: Flexible linear openings and closings. In: Mathematical Morphology and its Applications to Image and Signal Processing, pp. 109–118. Springer, New York (2000)
8. 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)
9. Del Moral, P.: Maslov Optimization Theory: Optimality versus Randomness. In: Kolokoltsov, Maslov (eds.) Idempotency Analysis and its Applications, Kluwer Publishers (1997)
10. Del Moral, P., Salut, G.: Random Particle Methods in (max, +)-optimization problems. In: Gunawardena (ed.) Idempotency, Cambridge University Press (1998)
11. Estrada, F.J., Fleet, D.J., Jepson, A.D.: Stochastic Image Denoising. In: Proc. of BMVC 2009 (2009)
12. Gondran, M., Minoux, M.: Graphes et Algorithmes. Eyrolles (1995)
13. Grazzini, J., Soille, P.: Edge-preserving smoothing using a similarity measure in adaptive geodesic neighbourhoods. Pattern Recognition 42(10), 2306–2316 (2009)
14. Heijmans, H.J.A.M.: Morphological image operators. Academic Press, Boston (1994)
15. Heijmans, H.J.A.M., van den Boomgaard, R.: Algebraic Framework for linear and morphological scale-spaces. Journal of Visual Communication and Image Representation 13(1-2), 269–301 (2002)
16. Heijmans, H., Buckley, M., Talbot, H.: Path openings and closings. Journal of Mathematical Imaging and Vision 22(2-3), 107–119 (2005)
17. Lerallut, R., Decencière, E., Meyer, F.: Image filtering using morphological amoebas. Image and Vision Computing 25(4), 395–404 (2007)
18. Maslov, V.: Méthodes opératorielles. Editions Mir (1987)
19. Maragos, P.: Slope Transforms: Theory and Application to Nonlinear Signal Processing. IEEE Trans. on Signal Processing 43(4), 864–877 (1995)
20. Meyer, F.: Grey-weighted, ultrametric and lexicographic distances. In: Mathematical Morphology: 40 Years On (Proc. of ISMM 2005). Computational Imaging and Vision, vol. 30, pp. 289–298. Springer (2005)
21. Stawiaski, J.: Optimal Path: Theory and Models for Vessel Segmentation. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 417–428. Springer, Heidelberg (2011)
22. Velasco-Forero, S., Angulo, J.: On Nonlocal Mathematical Morphology. In: Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 219–230. Springer, Heidelberg (2013)

Salience-Based Parabolic Structuring Functions

Vladimir Ćurić and Cris L. Luengo Hendriks

Centre for Image Analysis
Uppsala University and Swedish University of Agricultural Sciences, Uppsala, Sweden
{vlada,cris}@cb.uu.se

Abstract. It has been shown that the use of the salience map based on the salience distance transform can be useful for the construction of spatially adaptive structuring elements. In this paper, we propose salience-based parabolic structuring functions that are defined for a fixed, predefined spatial support, and have low computational complexity. In addition, we discuss how to properly define adjunct morphological operators using the new spatially adaptive structuring functions. It is also possible to obtain flat adaptive structuring elements by thresholding the salience-based parabolic structuring functions.

1 Introduction

Currently, one very active research area in mathematical morphology is on the construction of adaptive morphological operators, which are based on adaptive structuring elements that adjust their shape and/or size according to the local attributes of the image. A number of different methods for the construction of adaptive structuring elements have been proposed, where different image attributes were taken into account. To name a few: morphological amoebas [1], bilateral structuring functions [2], general adaptive neighbourhood image processing [3], gradient orientation fields [4], salience adaptive structuring elements [5].

We have recently proposed two different methods for adaptive structuring elements that are based on salience features in the image, in particular based on the salience map obtained by the salience distance transform of the edge strength of the input image. First, we have proposed salience adaptive structuring elements, where the structuring elements are based on path distances defined on the salience map [5]. This approach is similar to morphological amoebas [1] and geodesic adaptive neighbourhoods [6]. Both the shape and size of salience adaptive structuring elements change over the image, whereas the two aforementioned methods only change the shape of the structuring element. The second method [7] is based on predefined shapes of structuring elements where the size (the radius) of structuring elements is determined using a similar salience map. The first method provides structuring elements that are more flexible, and consequently the morphological operators have better properties than ones defined by the latter method. Nevertheless, with the later approach the structuring elements can be computed in linear time with respect to the number of pixels, N, in the image. This is not the case with the salience adaptive structuring elements,

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 183–194, 2013.

which are computable in $\mathcal{O}(Nr^2 \log r^2)$, where r is the radius of the structuring elements.

Taking into account that the use of salience features in the image is beneficial for the construction of adaptive structuring elements, in this paper we introduce a family of salience-based parabolic structuring functions. These functions are defined on a fixed spatial support. We also introduce flat adaptive structuring elements derived from salience-based parabolic structuring functions by a threshold. In addition, we discuss how to properly define adjunct morphological operators for spatially adaptive structuring functions.

2 Previous Work on Adaptive Mathematical Morphology

Let $D \subset \mathbb{Z}^n$ be the domain space of the image f, and $T = \mathbb{R} \cup \{-\infty, \infty\}$ be the range space representing the image values. Then, any image f can be represented by a function $f : D \to T$. Let $(\mathcal{L}, \leq)$ be a lattice defined for all grey valued functions with the domain D and the range T, where the partial ordering $\leq$ is defined as $f \leq g$ if and only if $f(x) \leq g(x), x \in D$.

Two basic morphological operators, an erosion $\varepsilon : \mathcal{L} \to \mathcal{L}$ and a dilation $\delta : \mathcal{L} \to \mathcal{L}$ for a function f can be defined, respectively, as

$$\varepsilon(f)(x) = \bigwedge_{y \in D} (f(x+y) - b(y)), \quad x \in D \tag{1}$$

and

$$\delta(f)(x) = \bigvee_{y \in D} (f(x-y) + b(y)), \quad x \in D \tag{2}$$

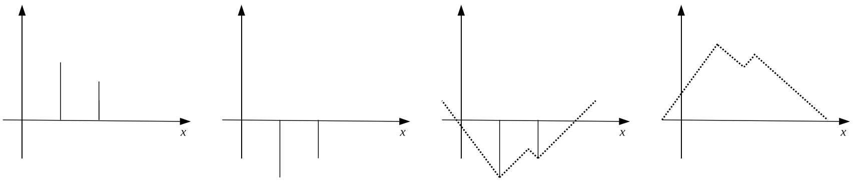
where $b : D \to T$ is a structuring function, $\bigwedge$ and $\bigvee$ denotes the infimum and supremum, respectively.

Structuring elements are actually flat structuring functions that have values 0 and $-\infty$, and they can be obtained by thresholding the corresponding structuring functions at some threshold value τ.

In traditional mathematical morphology (defined for non-adaptive structuring elements), the same structuring function $b(x)$ (structuring element B_x) is considered for all $x \in D$, i.e. for all pixels in the image. In adaptive mathematical morphology, the construction of structuring functions is dependent on the position in space D, i.e. morphological operators are adaptive with respect to the position in the considered space.

2.1 Adaptive Mathematical Morphology

One of the initial studies on adaptive structuring elements from a theoretical point of view was done by Serra [8]. Theoretical advances were also presented in two recent papers by Bouaynaya et al. [9,10], while Roerdink [11] discussed how to properly construct adjunct morphological operators (for instance, an erosion and its adjunct dilation) using adaptive structuring elements, and consequently



(a) NMS(f) (b) $-$NMS(f) (c) SDT (d) SM

Fig. 1. Computation of the salience map SM for a one dimensional function f

adaptive morphological openings and closings. Maragos and Vachier [12] recently presented a good overview on adaptive mathematical morphology.

A number of different methods for the construction of adaptive structuring elements, and consequently adaptive morphological operators, have been recently proposed. Methods for spatially adaptive structuring elements based on path distances are morphological amoebas [1], adaptive geodesic neighbourhoods [6], and salience adaptive structuring elements [5]. Adaptive structuring elements can also be based on gradient orientation fields [4], or may be determined by intensity information as a connected component that contains the origin of the structuring element [3]. The shape of structuring elements is adaptive for all the aforementioned methods, while the size is usually predefined, but can also be adaptive to the local image attributes, as is the case with the salience adaptive structuring elements.

Non-flat adaptive structuring elements, i.e. adaptive structuring functions, can be obtained in a predefined fixed support (shape), while the weights are defined using different image attributes. Then, adaptive structuring elements can be computed by thresholding the corresponding structuring functions. For instance, the two approaches that follow this idea are spatially variant bilateral functions [2] and nonlocal structuring functions [13].

2.2 Salience-Based Approaches for Adaptive Mathematical Morphology

Two methods for the construction of adaptive structuring elements based on salience features in the image have been recently proposed [5, 7]. These two methods are based on the salience map that was constructed using the salience distance transform (SDT) [14] by taking into account salience of the edges in the image. In particular, we have considered the salience map where important or salience features of the image are highly weighed. This salience map was constructed using SDT, which is a modified version of the distance transform, where instead of having binary image as input, any image attribute can be used, such as the edge strength, edge length, or curvature. Consequently, such a salience map contains the information about the spatial distance between the pixels, and also takes into account the salience of the edges in the image. Several different algorithms for the computation of the salience distance transform were proposed [14, 15]. We used a linear algorithm and took the gradient magnitude as the edge salience. The edge pixels were initialized with the negative values of their salience and the non-edge pixels were set to infinity [14]. Then, SDT was computed with the classical two-pass chamfering algorithm [16] using any type of distance propagation. In this paper, we used the Euclidean distance transform, but any other type of distance transform could be used.

The edges in the image are determined by using Canny's edge detector [17], where we use only the estimation of the gradient and non-maximal suppression (denoted as NMS(f), see Fig. 1a), and exclude hysteresis thresholding. The gradient of the input image is computed using Gaussian derivatives, with the standard deviation of the Gaussian filter set to $\sigma = 1$. Formally speaking, the salience map SM (Fig. 1d) can be written as

$$\mathrm{SM}(y) = \mathrm{Offset} + \bigvee_{x \in D} \Big(\mathrm{NMS}(f)(x) - \|x - y\| \Big),\ y \in D,$$

where $\mathrm{Offset} = \bigwedge_{y \in D} \bigvee_{x \in D} \Big(\mathrm{NMS}(f)(x) - \|x - y\| \Big)$ and $\|x - y\|$ is the Euclidean distance. The considered Offset is used in order to have positiveness of the salience map SM.

The main idea with the two proposed methods for adaptive structuring elements based on the salience map SM is that the structuring elements are smaller in size at points with larger values of SM, i.e., at the points where the edges are in the input image. Additionally, the structuring elements grow more away from the edge rather than in the direction close to the edge. In the first method [5], the salience adaptive structuring elements were proposed using path-based distances on the salience map SM, such that the cost c of the path between two adjacent pixels $(p, f(p))$ and $(q, f(q))$ is defined as $c(p, q) = \mathrm{SM}(p) + \mathrm{SM}(q)$. In addition, the size of the structuring elements is also determined by the SM. The second method [7] introduces structuring elements that have fixed and predefined shape, but their size is determined by the local minima and maxima in the salience map SM. Morphological operators defined with the first method have better properties than morphological operators defined with the second method, but, as mentioned in Section 1, their computational complexity is higher.

3 Salience-Based Parabolic Structuring Functions

Parabolic structuring functions are the only structuring functions that are separable and isotropic [18]. In addition, parabolic structuring functions are eigenfunctions of the slope transform [19]. In the context of adaptive mathematical morphology, spatially variant structuring functions based on bilateral filtering were recently introduced by Angulo [2] as

$$b_{\sigma_s, \sigma_i}(x) = - \left(\frac{\|x - y\|^2}{2\sigma_s^2} + \frac{|f_\rho(x) - f_\rho(y)|^2}{2\sigma_i^2} \right),$$

for all $x \in D$. The function f_ρ is a smoothed version of the input image f, and σ_s, σ_i are spatial and intensity parameters, respectively. These bilateral structuring functions are defined on the windowed (fixed) spatial support $D_S(x)$ around every point $x \in D$. In the aforementioned method and in one based on non-local means [13], the structuring functions were computed on the fixed spatial support, where the values of the structuring functions depend on image attributes. Note that the fixed spatial support of structuring functions can be a disk, an octagon, a rectangle, or any other shape.

We have been inspired by the aforementioned methods, and we propose spatially variant parabolic structuring functions based on the salience map SM that are defined on a fixed spatial support. In the two previous methods for adaptive structuring elements based on the salience map SM [5,7], the size of structuring elements decreases as the salience map SM increases and vice versa. To preserve the same property of structuring elements (structuring functions), we define a family of salience-based parabolic structuring functions centered at point $x \in D$ as

$$b_s^x(y) = -\big(\alpha(\mathrm{SM}(x), \mathrm{SM}(y)) + \beta\,\mathrm{SM}(x)\|x - y\|^2\big), y \in D_S(x), \tag{3}$$

where $\alpha(\mathrm{SM}(x), \mathrm{SM}(y)) \geq 0$ is a monotonically non-decreasing function, $\beta \geq 0$ is a constant, and $D_S(x) = \{x + z : \|z\| \leq R, x \in D\}$ ($\|\cdot\|$ is the Euclidean distance). This family satisfies the following property

$$\text{If } \ \mathrm{SM}(x_1) \leq \mathrm{SM}(x_2) \ \text{ then } \ b_s^{x_1}(x_1 + y) \geq b_s^{x_2}(x_2 + y), \ \text{ for all } \ x_1, x_2 \in D. \tag{4}$$

In plain English, property (4) says that the salience-based parabolic structuring functions are larger where the salience map SM is smaller (see Fig. 2a), and that the salience-based parabolic structuring functions are smaller if close to the edge (see Fig 2b). In other words, the parabola centered at point x_3 (Fig. 2b) is not symmetric and it decreases faster at the side closer to the higher salience, i.e. the parabola $b_s^{x_3}(y)$ is skewed away from the edges in the input image.

Note that the salience adaptive structuring elements can be described using the newly proposed salience-based parabolic structuring functions (3). If $\alpha(\mathrm{SM}(x), \mathrm{SM}(y)) = \mathrm{SM}(x) + \mathrm{SM}(y)$ and $\beta = 0$ for all $y \in D_S(x)$, then $b_s^x(y) = -(\mathrm{SM}(x) + \mathrm{SM}(y))$, which means that the salience-based parabolic structuring functions become the salience adaptive structuring elements if $D_S(x)$ is the 8-connected neighbourhood of a point $x \in D$.

Once spatially adaptive structuring functions are constructed for each point in the image, basic morphological operators can be defined using these structuring functions. It is straightforward to compute morphological erosion and dilation using formulas (1) and (2). Nevertheless, morphological operators defined in this way might not be adjunct morphological operators, and operators defined by their superpositions do not satisfy properties of morphological openings and closing. This fact seems to be overlooked in the literature. For instance, Salembier [13] discussed this issue, but only mentioned that the combinations of erosions and dilations will not provide opening and closing since they are not idempotent in the general case, which is correct. However, two important properties that often fail when computing erosion (1) and dilation (2) for spatially adaptive structuring functions are extensiveness for morphological closing and

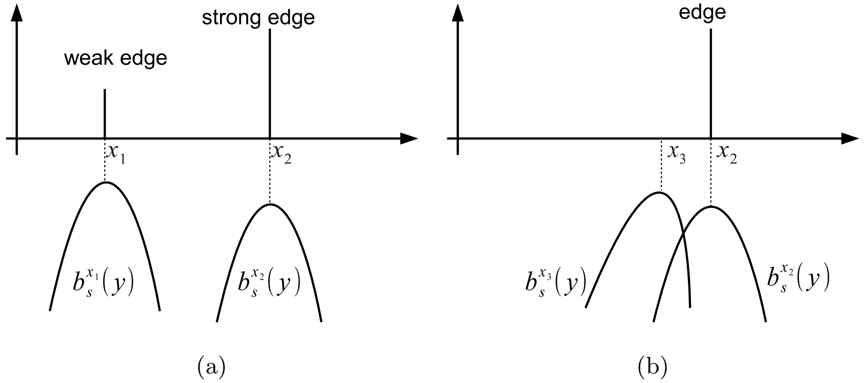


Fig. 2. Salience-based parabolic structuring functions in 1D. (a) Two structuring functions centered at edge points. (b) Two structuring functions where one is centered at the edge and one is close to the edge.

anti-extensiveness for morphological opening. Angulo [2] clearly mentioned the issue with adjunct morphological operators based on spatially-adaptive structuring functions, but does not explicitly deal with it and continues the discussion using counter-harmonic mean. Here, we present a way how to compute adjunct erosion and dilation using spatially adaptive structuring functions.

Two operators $\varepsilon : \mathcal{L} \to \mathcal{L}$ and $\delta : \mathcal{L} \to \mathcal{L}$ are adjunct operators if

$$\delta(f) \leq g \Leftrightarrow f \leq \varepsilon(g), \tag{5}$$

where $f, g : D \to T$. To properly construct adjunct morphological operators for spatially adaptive structuring functions, we define an erosion as

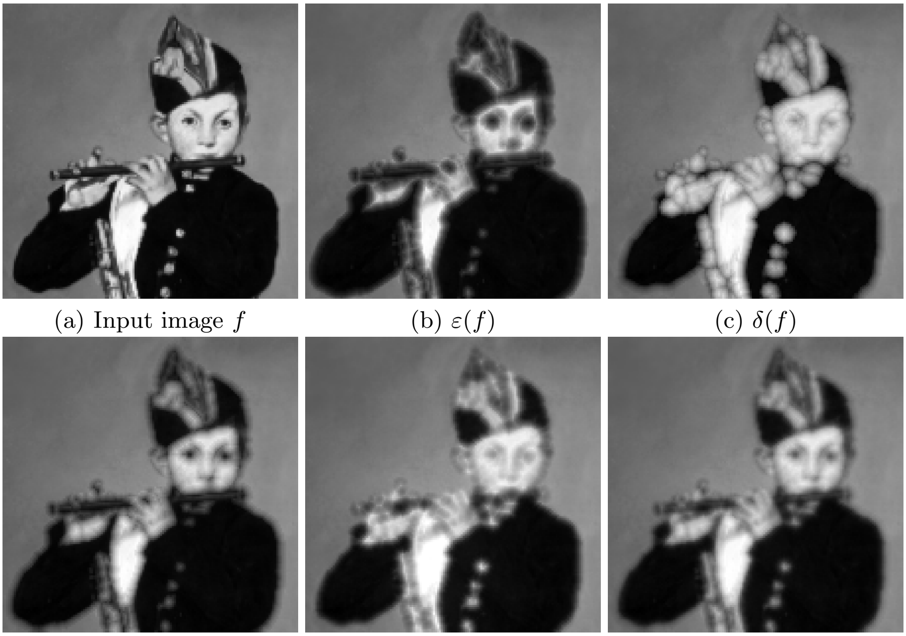
$$\varepsilon(f)(x) = \bigwedge_{y \in D_S(x)} (f(y) - b_s^x(y)),\ x \in D \tag{6}$$

and its adjunct dilation as

$$\delta(f)(x) = \bigvee_{y \in D_S(x)} (f(y) + b_s^y(x)),\ x \in D. \tag{7}$$

Note that structuring functions in (6) and (7) are only defined for points in the spatial support $D_S(x)$.

When morphological erosion and dilation are defined with (6) and (7), then morphological operators as superpositions of these operators can be defined as $\gamma(f) = \delta(\varepsilon(f))$ and $\phi(f) = \varepsilon(\delta(f))$. Note that these two operators γ and ϕ can only be considered as morphological opening and closing, respectively, if the same structuring functions are used for erosion and its adjunct dilation, i.e. the salience map SM is computed only once.



(a) Input image f (b) $\varepsilon(f)$ (c) $\delta(f)$

(d) $\gamma(f)$ (e) $\phi(f)$ (f) $(\gamma(f) + \phi(f))/2$

Fig. 3. Morphological operators for salience-based parabolic structuring functions with $\alpha = 0$, $\beta = 0.1$, $R = 10$

The following properties are satisfied for operators $\gamma : \mathcal{L} \to \mathcal{L}$ and $\phi : \mathcal{L} \to \mathcal{L}$, if and only if the same salience map SM is used for all computations:

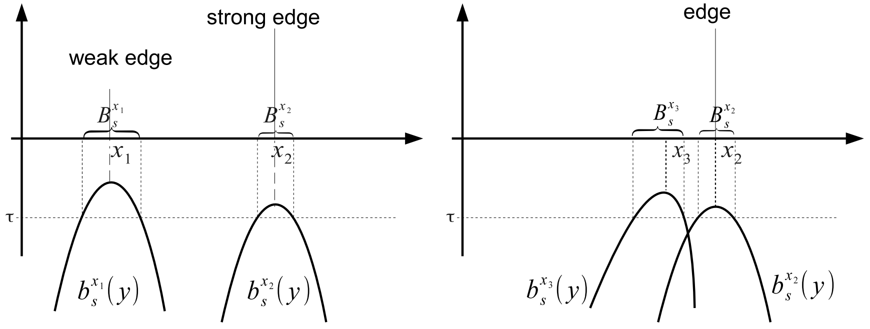
1. $(\phi(f^C))^C = \gamma(f)$, where C denotes the complement of the image with respect to space D (duality)
2. $\phi(f) < \phi(g)$ and $\gamma(f) < \gamma(g)$, if $f < g$ (increasingness)
3. $f < \phi(f)$ (extensiveness of ϕ)
4. $\gamma(f) < f$ (anti-extensiveness of γ)
5. $\gamma(\gamma(f)) = \gamma(f)$ and $\phi(\phi(f)) = \phi(f)$ (idempotency)

which implies that the two operators γ and ϕ are a morphological opening and closing, respectively.

The experimental results for the salience-based parabolic structuring functions are shown in Fig. 3. In this test we use $b_s^x(y) = -0.1\,\mathrm{SM}(x)\|x - y\|^2$, $y \in D_S(x)$, $x \in D$. It can be seen that presented morphological operators do mostly change the object in the image rather than the background of the object.

4 Flat Adaptive Structuring Elements Derived from Salience-Based Parabolic Structuring Functions

In this section, we focus on flat adaptive structuring elements. These structuring elements take only two values 0 and $-\infty$, and are obtained by thresholding the



(a) (b)

Fig. 4. Flat adaptive structuring elements derived from salience-based parabolic structuring functions at some threshold level τ for a 1D function

corresponding salience-based structuring functions at some threshold value τ (see Fig. 4),

$$B_s^x(y) = \begin{cases} 0, & b_s^x(y) \geq \tau \\ -\infty, & b_s^x(y) < \tau \end{cases}$$

where $x \in D$ and $y \in D_S(x)$. As explained in Section 3, if $\mathrm{SM}(x_1) \leq \mathrm{SM}(x_2)$ then $b_s^{x_1}(x_1 + y) \geq b_s^{x_2}(x_2 + y)$, which implies that $B_s^{x_2}$, translated to the origin, is a subset of $B_s^{x_1}$, translated to the origin.

Following the properties of the salience-based structuring functions, the structuring elements defined in this way are smaller close to stronger edges (Fig. 4a), and grow away from the edges rather than towards them (Fig. 4b).

Morphological erosion $\varepsilon : \mathcal{L} \to \mathcal{L}$ and dilation $\delta : \mathcal{L} \to \mathcal{L}$ for salience-based parabolic structuring elements that satisfy the adjunction property (5) can be defined [11], respectively, as

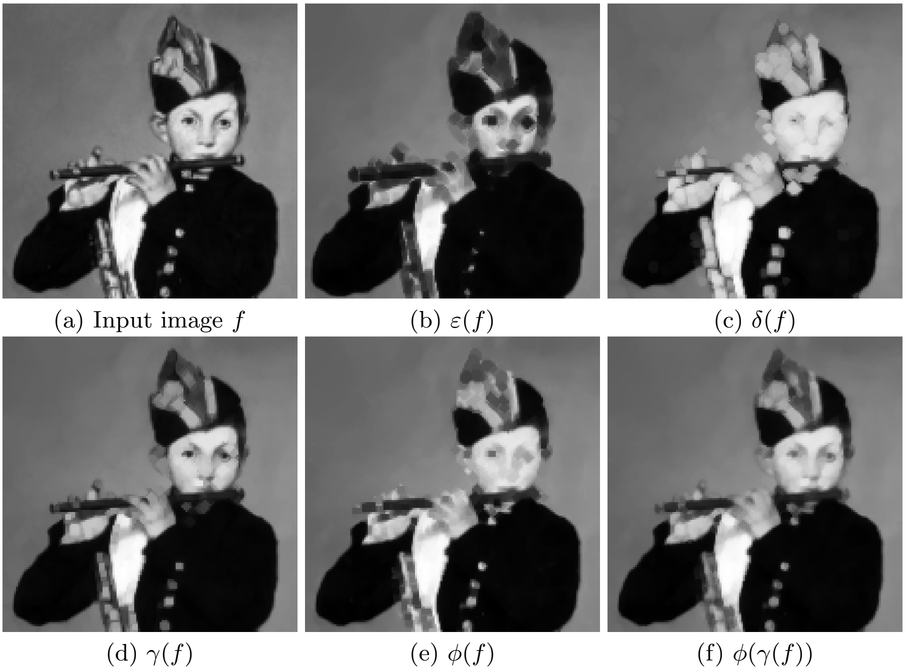
$$(\varepsilon(f))(x) = \bigwedge_{y \in B_s^x} f(y), \quad x \in D \tag{8}$$

$$(\delta(f))(x) = \bigvee_{y \in \breve{B}_s^x} f(y), \quad x \in D \tag{9}$$

where $\breve{B}_s^x$ is the reflected neighbourhood defined as

$$y \in B_s^x \Leftrightarrow x \in \breve{B}_s^y .$$

Accordingly, the corresponding morphological opening and closing can be defined by $\gamma(f) = \delta(\varepsilon(f))$ and $\phi(f) = \varepsilon(\delta(f))$, respectively. We use the following algorithm to compute adjunct morphological operators without computing the reflected neighbourhood of the flat adaptive structuring elements derived from salience-based parabolic structuring functions. The erosion can be computed by



(a) Input image f

(b) $\varepsilon(f)$

(c) $\delta(f)$

(d) $\gamma(f)$

(e) $\phi(f)$

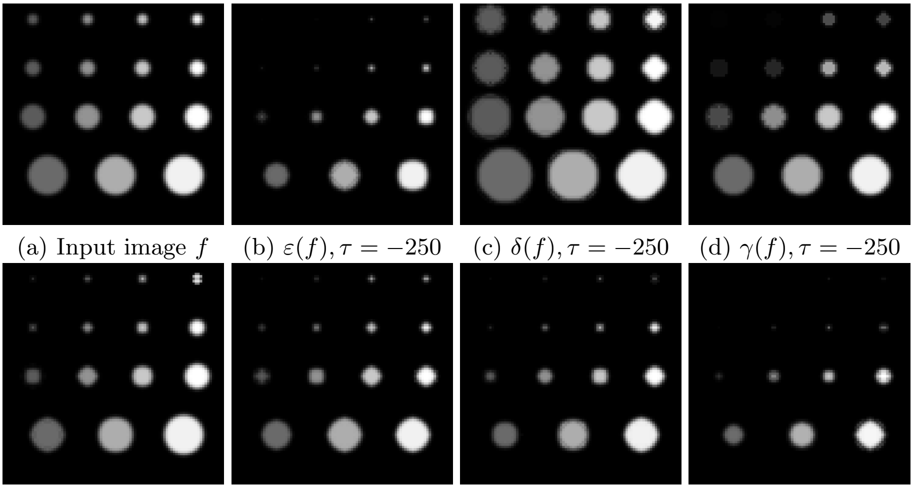
(f) $\phi(\gamma(f))$

Fig. 5. Morphological operators with flat adaptive structuring elements derived from salience-based parabolic structuring functions for $\alpha = 0$, $\beta = 1$, $R = 10$ and threshold $\tau = -350$

taking the infimum of the values over the structuring element, and its adjunct dilation is computed by [1]:

```
for each point $x \in D$ do
    compute $B_s^x$
    for each $y \in B_s^x$ do
        $\delta(y) = \max(f(x), \delta(y))$
    end for
end for
```

The important advantage of the proposed method with the fixed spatial support is that it uses distances that can be propagated in linear time (for instance, the Euclidean distance), while morphological amoebas or salience adaptive structuring elements are based on a geodesic distance, and have a relatively high computational complexity. The salience map SM can be efficiently computed in linear time with respect to the number of pixels in the image. Then, the salience-based parabolic structuring functions can be computed in $\mathcal{O}(\omega N)$, where ω is the size of the spatial support of the structuring functions. For the experiments in this paper, we used a disk with radius R as a spatial support D_S, which gives a time complexity of $\mathcal{O}(R^2 N)$. This computational complexity is less that the one



(a) Input image f (b) $\varepsilon(f), \tau = -250$ (c) $\delta(f), \tau = -250$ (d) $\gamma(f), \tau = -250$

(e) $\varepsilon(f), \tau = -100$ (f) $\varepsilon(f), \tau = -150$ (g) $\varepsilon(f), \tau = -200$ (h) $\varepsilon(f), \tau = -350$

Fig. 6. Morphological operators with flat adaptive structuring elements derived from salience-based parabolic structuring functions for $\alpha = 0$, $\beta = 1$, $R = 8$ and different thresholds τ

required for the salience adaptive structuring elements, $\mathcal{O}(Nr^2 \log r^2)$, where r is the radius of these adaptive structuring elements.

Two examples of the proposed adaptive structuring elements are depicted in Fig. 5 and Fig. 6. Note that morphological operators defined with the new adaptive structuring elements have different behaviour for the circles with different contrast, which is not the case with the traditional morphological operators with fixed structuring elements.

5 Summary and Conclusions

In this paper, we have introduced salience-based parabolic structuring functions that are defined on a fixed spatial support and depend on the salience map obtained from the salience distance transform of the input image. We have also shown that these adaptive structuring functions can be considered as a generalization of the salience adaptive structuring elements [5]. In addition, we proposed adaptive morphological operators with adaptive flat structuring elements that are obtained by thresholding the salience-based parabolic structuring functions.

Furthermore, we have used a proper way of computing morphological operators, in particular adjunct morphological dilation and erosion, using spatially-adaptive structuring functions. This issue seems to be overlooked in the literature, and we empirically show how to construct those operators using any spatially-adaptive structuring function.

The proposed adaptive morphological operators defined with the salience-based parabolic structuring functions (or their thresholded counterparts) have interesting properties, compared with both adaptive and non-adaptive morphological operators. One of the main advantages of the proposed method over previously introduced adaptive structuring elements is its low computational complexity.

The current work experimentally shows how to define proper adjunct morphological operators for spatially adaptive structuring functions, and this work should be pursued further from a theoretical point of view.

Acknowledgement. This work is financed by the Graduate School in Mathematics and Computing at Uppsala University, Sweden. Scientific support from Prof. Gunilla Borgefors is highly appreciated.

References

1. Lerallut, R., Decencière, E., Meyer, F.: Image filtering using morphological amoebas. Image and Vision Computing 25(4), 395–404 (2007)
2. Angulo, J.: Morphological Bilateral Filtering and Spatially-Variant Adaptive Structuring Functions. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 212–223. Springer, Heidelberg (2011)
3. Debayle, J., Pinoli, J.: Spatially Adaptive Morphological Image Filtering using Intrinsic Structuring Elements. Image Analysis and Stereology 24(3), 145–158 (2005)
4. Verdú-Monedero, R., Angulo, J., Serra, J.: Anisotropic morphological filters with spatially-variant structuring elements based on image-dependent gradient fields. IEEE Transactions on Image Processing 20(1), 200–212 (2011)
5. Ćurić, V., Luengo Hendriks, C.L., Borgefors, G.: Salience adaptive structuring elements. IEEE Journal of Selected Topics in Signal Processing 6(7), 809–819 (2012)
6. Grazzini, J., Soille, P.: Edge-preserving smoothing using a similarity measure in adaptive geodesic neighbourhoods. Pattern Recognition 42(10), 2306–2316 (2009)
7. Ćurić, V., Luengo Hendriks, C.L.: Adaptive structuring elements based on salience information. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2012. LNCS, vol. 7594, pp. 321–328. Springer, Heidelberg (2012)
8. Serra, J.: Image Analysis and Mathematical Morphology, vol. 2: Theoretical advances. Academic Press, New York (1988)
9. Bouaynaya, N., Charif-Chefchaouni, M., Schonfeld, D.: Theoretical Foundation of Spatially-Variant Mathematical Morphology Part I: Binary Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(5), 823–836 (2008)
10. Bouaynaya, N., Schonfeld, D.: Theoretical Foundation of Spatially-Variant Mathematical Morphology Part II: Gray-Level Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(5), 837–850 (2008)
11. Roerdink, J.B.T.M.: Adaptive and group invariance in mathematical morphology. In: Proc. of IEEE International Conference on Image Processing, pp. 2253–2256 (2009)
12. Maragos, P.A., Vachier, C.: Overview of adaptive morphology: Trends and perspectives. In: Proc. of IEEE International Conference on Image Processing, pp. 2241–2244 (2009)
13. Salembier, P.: Study on nonlocal morphology. In: Proc. of IEEE International Conference on Image Processing, pp. 2269–2272
14. Rosin, P., West, G.: Salience distance transforms. CVGIP: Graphical Models and Image Processing 57(6), 483–521 (1995)

15. Rosin, P.: A simple method for detecting salient regions. Pattern Recognition 42(11), 2363–2371 (2009)
16. Borgefors, G.: Distance transformations in digital images. Computer Vision, Graphics and Image Processing 34, 344–371 (1986)
17. Canny, J.: A Computational Approach To Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), 679–698 (1986)
18. van den Boomgaard, R.: Mathematical Morphology: Extensions Towards Computer Vision. PhD thesis, University of Amsterdam, Amsterdam (1992)
19. Dorst, L., van den Boomgaard, R.: Morphological Signal Processing and the Slope Transform. Signal Processing 38, 79–98 (1994)

Adaptive Morphologic Regularizations for Inverse Problems

Pulak Purkait and Bhabatosh Chanda

ECSU, Indian Statistical Institute

Abstract. Regularization is an well-known technique for obtaining stable solution of ill-posed inverse problems. In this paper we establish a key relationship among the regularization methods with edge-preserving noise filtering method which leads to an efficient adaptive regularization methods. We show experimentally the efficiency and superiority of the proposed regularization methods for some inverse problems, e.g. deblurring and super-resolution (SR) image reconstruction.

1 Introduction

Observed images of a scene are usually degraded by blurring due to atmospheric turbulence and inappropriate camera settings. The images are farther degraded by the various noises present in the environment and the system. Moreover, sometimes we end up with low-resolution (LR) image of the scene due to hardware limitations (e.g., sensor size) of a digital camera. Therefore, it is essential to get a sharp clean image from the noisy blurred image or to get a high-resolution (HR) image from multiple low-resolution image frames. These are called deblurring and super-resolution (SR) respectively and are classical inverse problems in image processing. In these inverse problems, the relation between the observed image Y and the desired sharp image X can be represented as [1–4]

$$Y = HX + \eta \tag{1.1}$$

where Y, X and η represent lexicographically ordered column vectors of the observed image, desired sharp image and the additive noise respectively. H is blurring matrix for deblurring problem. For SR problem, H embeds in itself geometric transformation (i.e. subpixel-shifts), downsampling and blurring [4].

The classical inverse problems (i.e., (1.1)) has been studied in both frequency and spatial domains extensively in last few decades. In frequency domain Wiener filter [5] is the most popular one that minimizes the mean square error. In spatial domain, the solution is obtained by solving an unconstrained optimization problem, such as

$$\hat{X} = \arg\min_X \{\frac{1}{2}\|HX - Y\|_2^2 + \mu\Upsilon(X)\} \tag{1.2}$$

where μ is the regularization parameter (also known as Lagrange multiplier) that controls the emphasis between the data error (first term) and the regularization (second term). $\Upsilon(X)$ mostly represent the smoothness criterion by means of energy in first or

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 195–206, 2013.

second order image derivative. However, in this work we concentrate only on regularization based spatial domain iterative algorithms which solve the minimization problem (1.2).

One of the most common regularization methods for inverse problem is Tikhonov regularization [6] using bounded variation (L_2 norm). It enforces smoothness in the reconstructed image, but at the same time loses some details (e.g., edges). An edge preserving regularization method is total variance (TV) using L_1 norm of the gradients or high-frequency wavelet components [2, 3, 7]. Farsiu *et al.* [8] employed bilateral total variation (BTV) and L_1 norm for both regularization and data fusion.

However, even though all these regularizations lead to stable solution, it is always interesting to explore different kind of regularization methods for betterment. So, developing an edge-preserving regularization method that can suppress noise in degraded images and the ringing artifacts evolved during reconstruction of image without sacrificing edges is yet to achieve. still an important task.

In this work, our contribution is multi-fold. First, we analyze the different regularization methods and show that they all emerge from the same concept involving low-pass filtering which is basically smoothness prior. We also analyze different types of smoothness kernels and found their commonalty. Based on these studies, we develop two adaptive kernels for low-pass morphologic filter using geodesic distance between the candidate pixel and its neighboring ones. Finally, combining all these, we devise two new regularization method as a prior for inverse image restoration problems.

2 Regularizer and Filtering for Inverse Problems

The objective of regularization is to ensure that the reconstructed image satisfies some quality criterion. From Bayesian point of view, regularization method corresponds to imposing certain prior in estimating image. For example, regularization may ensure smoothness in the reconstructed image or may improve the rate of convergence of iterative reconstruction algorithm. Here our intention is to develop a regularization method for estimating the output image with sharp edges and textures while suppressing the noise as well as undesired artifacts.

2.1 Regularization Techniques

To obtain a stable solution, suppose a specific regularization operator Υ is imposed on the estimated sharp image X. The regularization term $\Upsilon(X)$ (1.2) incorporates prior knowledge of the desired solution, e.g., degree of smoothness. A commonly used regularization term, viz, the Tikhonov cost function [6, 9–13] (sometimes called bounded variation (BV)), is given by

$$\Upsilon(X) = \|\Gamma X\|_2 \tag{2.1}$$

where Γ is usually a high-pass operator such as derivative or Laplacian. Since high-pass operators capture edges and noise with equal emphasis, minimization of $\|\Gamma X\|_2$

leads to suppression of noise along with blurring of edges. An early edge preserving regularization for inverse problems is total variation (TV) method [2, 3, 7] that penalizes the total amount of change in the image as measured by the L_1 norm of the magnitude of the gradient, i.e.,

$$\Upsilon(X) = |\nabla X|_1 \tag{2.2}$$

Bilateral total variation (BTV) [8] combines the total variation and the bilateral filter [14, 15]. This is extension of TV regularization and is given by

$$\Upsilon(X) = \sum_{l=-w}^{w} \sum_{m=-w}^{w} \beta^{|m|+|l|} |X - S_x^l S_y^m X|_1 \tag{2.3}$$

where S_x^l and S_y^m are shift operator matrices to represent l and m pixel shift in horizontal and vertical directions, respectively, such that $l+m \geq 0$. The term w is referred to as the window size and β $(0 < \beta < 1)$ is the weighting coefficient.

Purkait *et al.* [4] proposed the regularization function based on multi-scale morphology as:

$$\Upsilon(X) = \sum_{s=1}^{S} \alpha^s \mathbf{1}^t [C_s(X) - O_s(X)] \tag{2.4}$$

where $\mathbf{1}$ is a column vector consisting of all 1's and α is the weighting coefficient. $C_s(X) = (X \oplus sB) \ominus sB$ and $O_s(X) = (X \ominus sB) \oplus sB$ are morphological closing and opening operators with disk structuring element sB of size s. Clearly, regularization operators (2.2)-(2.4) are based on some high-pass operators. Ideal regularizer is expected to capture only noise present in the image so that minimizing $\Upsilon(X)$ leads to suppression of noise while keeping sharp edges unaltered. Among above regularizers, the morphologic regularizer (2.4) is the closest to the ideal, as the open and close filters remove noise smaller than SE and preserve edges upto the curvature of SE.

2.2 Regularizer and Smoothing Kernel

Smoothing a data-set mainly creates an approximating function that captures the underlying pattern in the data, while leaving out noise or other fine-scale structures. Traditionally, people use symmetric linear filter (e.g. Gaussian filtering) or nonlinear filter (e.g. median filter). The regularization techniques described in previous section may be viewed as a *prior* which suggests that the image is composed mostly of smooth regions. That means the reconstructed image would be mostly unaffected by low pass filtering with a constraint that it satisfies the forward image generating process (data error term of (1.2)). This constraint takes care of generating high frequency components due to edges.

Let $K = (\alpha_{lm} | l, m = -w : w)$ be the kernel of linear low pass spatial filter. For average filter $\alpha_{lm} = 1/S$, and for Gaussian filter $\alpha_{lm} = \frac{1}{S} \exp^{-\frac{l^2+m^2}{2\sigma^2}}$, where S is the normalizing constant so that $\sum_{l,m} \alpha_{lm} = 1$. Suppose W denotes an index set incorporates an window around a pixel. As discussed in the last section, an ideal image prior would be invariant to low pass filtering that can remove noise and keep all the image details. Then the regularization term may be defined as

$$\Upsilon(X) = \|X - KX\|_p \tag{2.5}$$

$$= \|X - \sum_{l,m\in W} \alpha_{lm} S_x^l S_y^m X\|_p \tag{2.6}$$

$$= \| \sum_{l,m\in W} \alpha_{lm}(X - S_x^l S_y^m X)\|_p \text{ (since } \sum_{l,m\in W} \alpha_{lm} = 1) \tag{2.7}$$

$$\leq \sum_{l,m\in W} \alpha_{lm}\|X - S_x^l S_y^m X\|_p \text{ (Jensen's Inequality), for } p > 1 \tag{2.8}$$

under minimization of $\Upsilon(X)$ [note that (1.2)], (2.8) is similar to (2.3). For average filtering ($\alpha_{lm} = 1/S$) with W as 4-neighborhood, $\Upsilon(X)$ becomes

$$\Upsilon(X) = \sum_{l,m\in W} \alpha_{lm}\|X - S_x^l S_y^m X\|_p = 2\|(\nabla_x + \nabla_y)X\|_p = \|\nabla X\|_p \tag{2.9}$$

which is similar to (2.2) for $p = 1$. An advantage of considering $\|\nabla X\|_1$ or $\sum_{l,m\in W} \alpha_{lm}$ $\|X - S_x^l S_y^m X\|_1$ as regularization term is that it essentially minimizes $\|X - KX\|_p$ which is our smoothness prior. i.e. minimization of eqs. (2.2)-(2.4) try to generate an image which is invariant to some smoothing operators and the data term of (1.2) is responsible to generate high frequency components of the image. If the data-term doesn't present then ideally it would produce a smooth flat image. Now, if $\Upsilon(X) = \|(\nabla_x + \nabla_y)X\|_p$, we end up with well-known BV (Bounded variation) and TV (Total variation) regularization for $p = 2$ and 1 respectively. Moreover, if we consider a larger window, $\Upsilon(X) = \sum_{l,m\in W} \alpha_{lm}\|X - S_x^l S_y^m X\|_p$, becomes BTV regularization for $p = 1$ and $\alpha_{lm} = \beta^{|m|+|l|}$ with $\beta \in (0,1)$. Note that $\Upsilon(X)$ basically measures energy in high-frequency components. Hence, an attractive alternate regularizer could be develop using spatial filter.

Spatial Image Filtering Techniques. Most of the traditional filters use a symmetric non-adaptive kernel. Those filters are able to remove noise and artifacts present in the image, but also blur the edges and wipe out textures. Tomasi *et al.* [15] used adaptive kernel for edge preserving image smoothing, known as bilateral filtering, given by

$$\alpha_{ij,\,lm} == \exp^{-\{\frac{l^2+m^2+\lambda(X(i+l,j+m)-X(i,j))^2}{2\sigma_e^2}\}} = \exp^{-\{\frac{D^2_{ij,\,lm}}{2\sigma_e^2}\}} \tag{2.10}$$

where σ_e is the scale parameter and $X(i,j)$ is the intensity value at $(i,j)^{th}$ pixels. λ controls the emphasis between spatial and spectral difference. This kernel may also be viewed as the Gaussian of Euclidean distance $D_{ij,\,lm}$ between value of the pixels [Fig. 1]. It is still one of the state-of-the-art linear methods for edge preserving smoothing. Various nonlinear edge preserving morphological smoothing algorithms are also available in the literature [16–19]. Lerallut et al. [16] define adaptive morphological operators, called amoeba filter, where shape of the structuring element varies from pixel to pixel, determined by thresholding the geodesic distance $D_{ij,\,lm}$ of the pixel

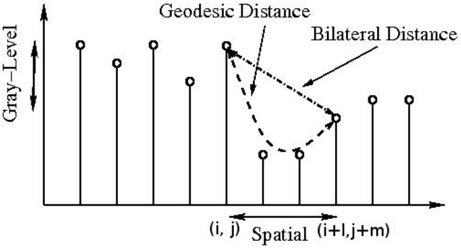


Fig. 1. Illustrates the bilateral and geodesic distances between points

$(i+l, j+m)$ from the center pixel (i, j) [Fig. 1]. Therefore it is clear that regularization methods used in the inverse problems (2.8) evolve from low-pass filtering. Hence, a better regularizer can be developed from more efficient adaptive nonlinear low-pass filters. Based on these observations we develop two geodesic distance based regularization methods for the inverse problem.

3 Proposed Morphologic Regularization

We define more general adaptive regularization as

$$\Upsilon(X) = \sum_{l,m \in W} \alpha_{lm} |\nabla^{lm}| \tag{3.1}$$

where α_{lm} is an adaptive kernel that varies from pixel to pixel. $|\nabla^{lm}| = \|X - S_x^l S_y^m X\|_p$ is p^{th} norm of the difference between the original image X and shifted one $S_x^l S_y^m X$ or $|\nabla^{lm}| = [C_s(X) - O_s(X)]$ is the norm of difference between morphologic closing and opening. In essence, a regularization term has two components: a derivative operator ∇^{lm} and a smoothing kernel α_{lm}. The elements of smoothing kernel α_{lm} decreases with distance from the candidate pixel (i, j). Morphologically viewed, an image may be considered as a topographic surface with (i, j) as spatial location and $X(i, j)$ as altitude. The geodesic distance in an image between pixels (i, j) and $(i+l, j+m)$ is defined as the length of the shortest path between them along the topographic surface. Note that BTV considers city-block distance (2.3) between pixel locations while bilateral filtering (2.10) considers Euclidean distance between two points on the topographic surface as discussed in the last section.

Topographic Morphology Based Regularization

Based on the above study on the kernels, we can choose an efficient adaptive kernel in the following form:

$$\alpha_{ij,\,lm} = \frac{1}{S_{ij}} \exp\{-\frac{D^p_{ij,\,lm}}{2\omega^p}\} \tag{3.2}$$

where $D^p_{ij,\,lm}$ is the geodesic distance of the pixels (i,j) and $(i+l,j+m)$ within W. S_{ij} is the normalizing factor of the kernel at location (i,j). Based on the above geodesic kernel $\alpha_{ij,\,lm}$, We propose a geodesic regularizer as follows:

$$\Upsilon(X) = \sum_{l,m\in W} \alpha_{lm}|X - S_x^l S_y^m X| \tag{3.3}$$

Morphologic Filter Based Regularization

Moreover, we use adaptive morphological operators (2.4) by choosing adaptive structuring element A adaptive instead of a flat one and is defined as

$$a_{ij,\,lm} = \kappa(\alpha_{ij,\,lm} - 1), \tag{3.4}$$

where κ is a constant and is fixed over all the pixels. Accordingly the adaptive morphological dilation is defined as:

$$(X \oplus A)(i,j) = \sup_{lm\in W} \{X(i-l,j-m) + a_{ij,lm}\}. \tag{3.5}$$

And adaptive morphological erosion is defined as:

$$(X \ominus A)(i,j) = \inf_{lm\in W} \{X(i+l,j+m) - a_{ij,lm}\} \tag{3.6}$$

Based on the above adaptive morphological operators, we propose adaptive morphologic regularizer as:

$$\Upsilon(X) = \sum_{l,m\in W} \alpha_{lm}[C_a(X) - O_a(X)] \tag{3.7}$$

where $C_a(X)$ and $O_a(X)$ are adaptive closing and opening operators.

The pixel-wise value due to different regularization terms (proposed as well as state-of-the-arts) for a noisy image are displayed in Fig. 2. Final regularization term is nothing but the aggregate of their magnitude, in short, energy. Image restoration algorithms try to minimize this energy along with the data error term (1.2). In TV and BTV regularization, edge information is quite significant. Thus minimization of these two regularizer leads to weakening or blurring the edges. It is clear from the image that proposed geodesic and adaptive morphologic regularization term extract more of noise than edges of the image. So in these two regularization terms noise energy is far more dominant than edge energy. In these cases, minimization of regularizers leads to suppression of noise leaving edge information more or less intact. This characteristics are also reflected in the quality of the restored image as will be seen in the experimental results. Finally, implementation of such regularizers require efficient computation of geodesic distance.

Geodesic Distance Computation

There are two main kinds of algorithms that exist in the literature for computing Geodesic distance: raster-scan and wave-front propagation. Raster-scan algorithms are based on kernel operations applied sequentially over the image in multiple passes [20].

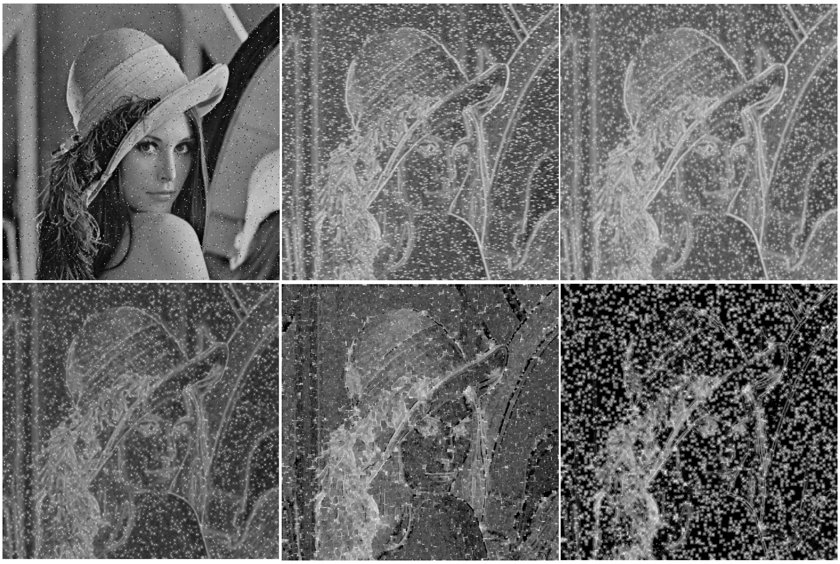


Fig. 2. Illustrates different regularization methods. Top row: noisy input image with salt and paper noise (2%), TV regularization term (2.2) and BTV regularization (2.3). Bottom row: deblurred image using geodesic regularization (3.3), Morphologic regularization (2.4), and Adaptive morphologic regularization (3.7) method. (Quantitative comparisons are in text).

Those methods are the extensions of Dijkstra algorithm for two dimensional grids. Wave-front algorithms such as Fast Marching Method (FMM) [21], on the other hand, are based on the iterative propagation of a pixel front with certain velocity. Geodesic versions of both kinds of algorithms may be found in [22] and [23].

Most of the existing geodesic distance computation techniques are developed for generating a distance transformation map for a gray-level image over a binary image. Whereas, in the present work, we need to compute the geodesic distance of every pixel within W from the central pixel. So here, we develop an efficient two-pass iterative raster-scan algorithm which is an extension of Dijkstra algorithm. Let $\Pi(ij,lm) = \{(i,j) = x_1, x_2, \ldots, (i+l, j+m) = x_n\}$ be a path in a connected domain between the pixels x_k and x_{k+1}. Then the geodesic distance between them is defined as:

$$D_{ij,\,lm} = \min_{\Pi}\left[\sum_{k=1}^{n-1}\{1+\gamma|X(x_i)-X(x_{i+1})|\}\right] \tag{3.8}$$

The factor γ weights the contribution of the intensity difference against the spatial distances. Equation (3.8) generalizes the conventional spatial distance which may be obtained by setting $\gamma = 0$.

We define derivative operator

$$\nabla_{ij}^{lm} = |X(i,j) - X(i+l, j+m)|_1, \text{ where } l,m \in \{-1,0,1\} \tag{3.9}$$

Then $\gamma\nabla_{ij}^{lm} + s$ is the geodesic distance of the $(i,j)^{th}$ pixel from $(l,m)^{th}$ neighboring pixel, where s represents the cost of unit step in spatial direction. We take $s = 1$ for

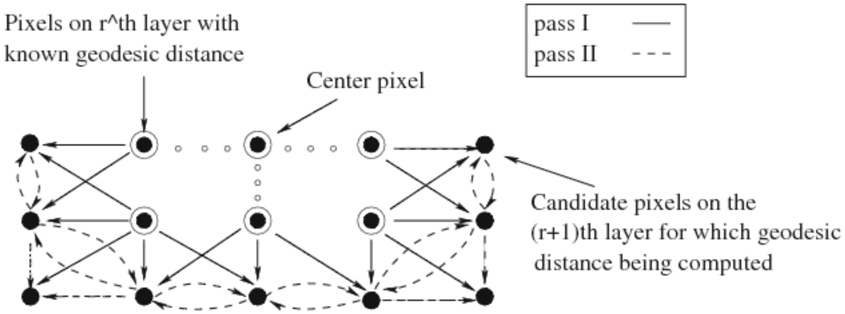

Pixels on r^th layer with known geodesic distance
Center pixel
pass I ——
pass II - - -
Candidate pixels on the (r+1)th layer for which geodesic distance being computed

Fig. 3. Flow chart of computation of geodesic distance of a pixel to its neighboring pixels

horizontal and vertical neighboring pixels and $s = \sqrt{2}$ for diagonal ones. The parameter γ controls the emphasis between spatial and spectral distances. Let Geo_{lm} represents geodesic distance from current $(i, j)^{th}$ pixel to $(i+l, j+m)^{th}$ pixel. We describe the computation of geodesic distance in concise algorithmic format as follows. Propagation of geodesic distance information is also illustrated in Fig. 3.

Algorithm

For central pixel (i, j), initialize $r = 0$, $Geo_{ij} = 0$
While r < w iterate Pass I and Pass II.

- **Pass I** : Compute interim geodesic distances of the pixel (l,m) on the $(r+1)^{th}$ layer from the central pixel along the possible shortest paths through the inner r^{th} layer.
  - Let $N_{lm}^{(r)}$ be the neighborhood of (l,m) consisting of adjacent pixels on the r^{th} layer as shown in figure 3 and consider it as a flat adaptive structuring element at (l,m).
  - Interim geodesic distance of pixel (l,m) from the center of the window by eroding with $N_{lm}^{(r)}$ as

$$Geo_{lm}^{tmp} = \min_{(u,v)\in N_{lm}^{r}} \{Geo_{uv} + \gamma \nabla_{lm}^{uv} + s\}$$

- **Pass II** : Compute actual geodesic distance of the pixel (l,m) on the $(r+1)^{th}$ layer using interim geodesic distances from Pass I.
  - Let $N_{lm}^{(r+1)}$ be the neighborhood consisting of adjacent pixels in the $(r+1)^{th}$ layer as shown in figure 3 and consider it as a flat adaptive SE at (l,m).
  - Actual geodesic distance of pixel (l,m) from the center of the window by eroding with N_{lm}^{r+1} as

$$Geo_{lm} = \min\left[Geo_{lm}^{tmp},\ \min_{(u,v)\in N_{lm}^{(r+1)}} \{Geo_{uv}^{tmp} + \gamma \nabla_{lm}^{uv} + s\}\right]$$

Now, we can compute kernel matrix $K = [\alpha_{ij,\,lm}]$ as in (3.2) with $D_{ij,\,lm} = Geo_{lm}$.

We solve (1.2) with the proposed adaptive regularization (3.1) by gradient descent technique. The non-differentiability and nonlinearity are handled by appropriate shrinkage operators [4]. The detail of optimization method is not included in this paper.

4 Experimental Results and Discussion

In this section we study the performance of the proposed regularization along with some state-of-the-arts regularization methods for two classical inverse problems: (i) Deblurring and (ii) Super Resolution.

Experimental Setting. For quantitative evaluation purpose, we have applied each algorithm on different sets of images with different parameters and the best result of each algorithm is chosen as the output of the algorithm. To compute geodesic kernel as described in (3.2), we have chosen $\omega = 5$ and $\gamma = 0.5$. For adaptive structuring element computation, we have chosen $\kappa = 60$ for gray-level range [0-255]. Except κ, other parameters do not have much critical effect on the performance of the regularizer and can be chosen from a wide range. We choose 512×512 gray-scale Lena image as test image with Gaussian blurring kernel ($\sigma_b = 2.5$) plus the Gaussian noise ($\sigma_a = 5$).

Experiment I: Deblurring. We synthesize a noisy blur image and then reconstruct the sharp image from it using the proposed regularization method as well as the existing regularization methods. Experimental results are shown in figure 4. It is seen that the morphological regularization yields better quality reconstructed images compared to other regularization methods. Experimental results on figure 4 provides the quantitative measure of quality as follows. The proposed geodesic regularization (3.3) (PSNR = 29.97dB, SSIM = 0.9228), proposed Adaptive morphologic regularization (3.7) method (PSNR = 30.10dB, SSIM = 0.9251), Rudin *et al.* [3] (PSNR = 29.93dB, SSIM = 0.9224), Farsiu *et al.* [8] (PSNR = 29.95dB, SSIM = 0.9224), and Purkait *et al.* [4] (PSNR = 30.05dB, SSIM = 0.9248) almost equally. However, though marginally the proposed regularizers perform best among the rest.

We did an experiment for adaptive morphological operators (3.4), with varying hight parameter κ of the structuring element. Then we plot the quantitative measures in Fig. 5. $\kappa = 0$ gives flat non-adaptive structuring element and in this case propose regularization method is identical with non-adaptive morphological regularization. As value of κ increases, quantitative measures also increases and after a certain value fo κ those start decreasing. It is clear from the plots that $\kappa = [50 - 70]$ gives good result in terms of quantitative measures.

Experiment II: Super Resolution Reconstruction. We synthesize 10 noise-free LR images down-sampled by resolution factor 5, and having different sub-pixel shifts. Then reconstruct HR image from these 10 LR images using the proposed regularization method as well as some existing regularization methods. Experimental results are shown in figure 6. It is seen that the proposed morphological regularization yields better quality SR reconstructed images, though not significantly very high, compared to other regularization methods. Experimental result on figure 6 also provides the quantitative

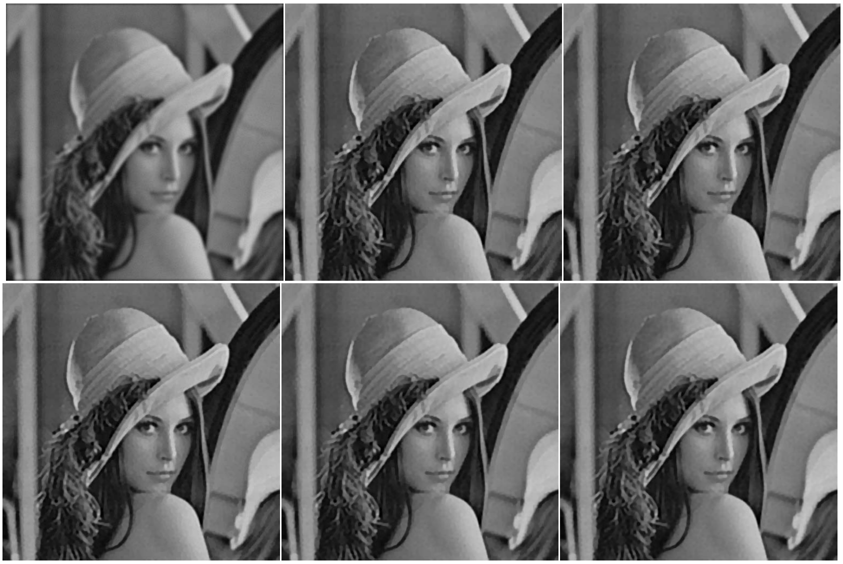


Fig. 4. Illustrates results of deblurring technique with various regularization methods. Top row: noisy and blurred input image, TV regularization [3] and BTV regularization [8]. Bottom row: deblurred image using geodesic regularization, Morphologic regularization [4], and Adaptive morphologic method. (Quantitative comparisons are in text).

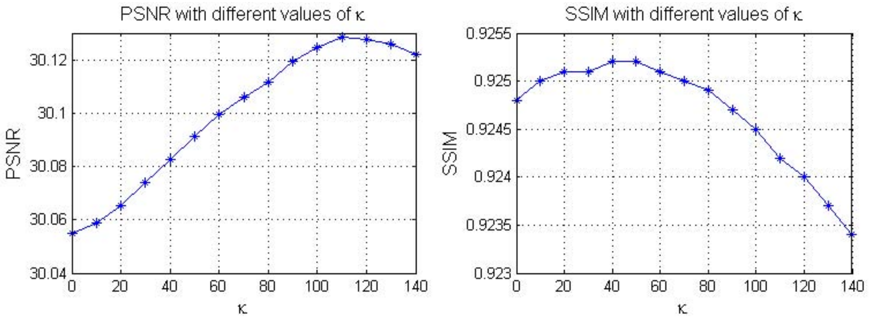


Fig. 5. Illustrates results of deblurring technique by proposed adaptive regularization method with different values of κ

measure of quality of proposed geodesic regularization (3.3) (PSNR = 29.48dB, SSIM = 0.9082), adaptive morphologic regularization (3.7) method (PSNR = 29.59dB, SSIM = 0.9107), Rudin *et al.* [3] (PSNR = 29.43dB, SSIM = 0.9077), Farsiu *et al.* [8] (PSNR = 29.47dB, SSIM = 0.9080), and of Purkait *et al.* [4] (PSNR = 29.55dB, SSIM = 0.9106). They perform equally well. However, though marginally, the proposed regularization methods perform better than others.

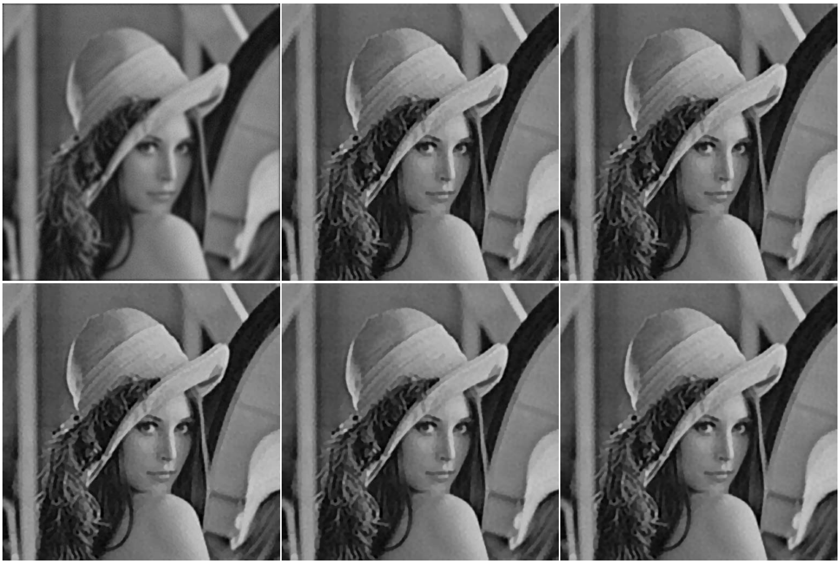


Fig. 6. Illustrates results of various SR image reconstruction techniques on Lena image. Top row : SR image using Bicubic interpolation, TV regularization [3] and BTV regularization [8]. Bottom row : SR reconstructed image using geodesic regularization, Morphologic regularization [4], and Adaptive morphologic method. (Quantitative comparisons are in text).

5 Conclusion

In this paper, we have proposed two geodesic kernel based regularization for a general inverse problem of image reconstruction. We analyze different regularization methods used in this domain and found that they come from the same concept of edge preserving smoother. We propose a new robust adaptive geodesic regularization and adaptive morphologic regularization methods that can suppress the noise more efficiently while preserving the edges. Our experimental section shows that it works quite well, in fact better than existing techniques.

The adaptive regularization method proposed here are tested for two classic inverse problems, viz, deblurring and SR reconstruction problem, but one can easily extend this work to other applications as well.

References

1. Chang, S.G., Yu, B., Vetterli, M.: Adaptive wavelet thresholding for image denoising and compression. IEEE Trans. Image Process. 9, 1532–1546 (2000)
2. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. of Imaging Sciences 2, 183–202 (2006)
3. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)

4. Purkait, P., Chanda, B.: Super resolution image reconstruction through bregman iteration using morphologic regularization. IEEE Transactions on Image Processing 21, 4029–4039 (2012)
5. Widrow, B., Stearns, S.: Adaptive signal processing, vol. 491, p. 1. Prentice-Hall, Inc., Englewood Cliffs (1985)
6. Zhang, X., Lam, E.Y., Wu, E.X., Wong, K.K.: Application of tikhonov regularization to super-resolution reconstruction of brain mri image. Medical Imaging and Informatics 49, 51–56 (2008)
7. Hansen, P.C., Nagy, J.G., OLeary, D.P.: Deblurring images. Fundamentals of Algorithms, vol. 3, pp. 291–294. Society for Industrial and Applied Mathematics (SIAM) (2006)
8. Farsiu, S., Robinson, M.D., Elad, M., Milanfar, P.: Fast and robust multiframe super-resolution. IEEE Transactions on Image Processing 13, 1327–1344 (2004)
9. Chan, T.F., Wong, C.K.T.: Multichannel image deconvolution by total variation regularization. In: Proceedings of SPIE, San Diego, CA, USA, pp. 358–366 (1997)
10. Li, X., Hu, Y., Gao, X., Tao, D., Ning, B.: A multi-frame image super-resolution method. Signal Processing 90, 405–414 (2010)
11. Nguyen, N., Milanfar, P., Golub, G.: Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement. IEEE Transactions on Image Processing 10, 1299–1308 (2001)
12. Nguyen, N., Milanfar, P., Golub, G.H.: A computationally efficient image super resolution algorithm. IEEE Transaction on Image Processing 10, 573–583 (2001)
13. Elad, M., Feuer, A.: Restoration of a single super-resolution image from several blurred, noisy, and under-sampled measured images. IEEE Transactions on Image Processing 6, 1646–1658 (1997)
14. Elad, M.: On the bilateral filter and ways to improve it. IEEE Transaction on Image Processing 11, 1141–1151 (2002)
15. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: IEEE International Conference on Computer Vision, New Delhi, India, pp. 836–846 (1998)
16. Lerallut, R., Decencière, E., Meyer, F.: Image filtering using morphological amoebas. Image and Vision Computing 25, 395–404 (2007)
17. Cheng, F., Venetsanopoulos, A.: Adaptive morphological operators, fast algorithms and their applications. Pattern Recognition 33, 917–933 (2000)
18. Soille, P.: Generalized geodesy via geodesic time. Pattern Recognition Letters 15, 1235–1240 (1994)
19. Debayle, J., Pinoli, J.C.: General adaptive neighborhood image processing. J. Math. Imaging Vis. 25, 267–284 (2006)
20. Borgefors, G.: Distance transformations in digital images. Comput. Vision Graph. Image Process. 34, 344–371 (1986)
21. Sethian, J.A.: Fast marching methods. SIAM Review 41, 199–235 (1999)
22. Toivanen, J.P.: New geodosic distance transforms for gray-scale images. Pattern Recognition Letters 17, 437–450 (1996)
23. Yatziv, L., Bartesaghi, A., Sapiro, G.: O(n) implementation of the fast marching algorithm. Journal of Computational Physics 212, 393–399 (2006)

Attribute Controlled Reconstruction and Adaptive Mathematical Morphology

Andrés Serna and Beatriz Marcotegui

Mines ParisTech
CMM - Centre de Morphologie Mathématique
Mathématiques et Systèmes
35 rue St Honoré 77305, Fontainebleau–CEDEX, France
{andres.serna_morales,beatriz.marcotegui}@mines-paristech.fr

Abstract. In this paper we present a reconstruction method controlled by the evolution of attributes. The process begins from a marker, propagated over increasing quasi–flat zones. The evolution of several increasing and non–increasing attributes is studied in order to select the appropriate region. Additionally, the combination of attributes can be used in a straightforward way.

To demonstrate the performance of our method, three applications are presented. Firstly, our method successfully segments connected objects in range images. Secondly, input–adaptive structuring elements (SE) are defined computing the controlled propagation for each pixel on a pilot image. Finally, input–adaptive SE are used to assess shape features on the image.

Our approach is multi–scale and auto–dual. Compared with other methods, it is based on a given attribute but does not require a size parameter in order to determine appropriate regions. It is useful to extract objects of a given shape. Additionally, our reconstruction is a connected operator since quasi–flat zones do not create new contours on the image.

Keywords: mathematical morphology, controlled reconstruction, connected operators, adaptive SE, quasi–flat zones, attribute evolution.

1 Introduction

Local operators constitute powerful techniques in digital image processing. They are based on the neighborhood of each pixel, defined by a kernel. In general, a kernel is a ball of radius r centered at the point to be processed. In the digital case, the kernel is reduced to the definition of a local neighborhood describing the connections between adjacent pixels. In Mathematical Morphology (MM), these kernels are called structuring elements (SE) and they are the base of sophisticated nonlinear techniques for filtering, feature extraction, detection and segmentation [1,2].

In practice, square SE are preferred. However, several works remark the usefulness and necessity of adapting algorithms according to intrinsic variability and

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 207–218, 2013.

a priori knowledge of the image [3]. Adaptive SE are elegant processing techniques using non-fixed kernels. Such operators, firstly introduced by Gordon and Rangayyan [4], vary their shape over the whole image taking into account local image features. Serra [2] called them *structuring functions* and defined erosion and dilation with spatially-varying SE.

In the literature, several works have been carried out with the aim of using image information in order to locally adapt SE shape and size. An overview on adaptive MM can be found in [3]. Most works focus on filters that privilege smoothing in homogeneous regions while preserving edges as well as possible. With this idea, Perona and Malik [5] proposed anisotropic filters that inhibit diffusion through strong gradients.

One of the first works using adaptive SE is due to Beucher [6]. He developed a traffic control application where the SE size depended on the perspective and varied linearly with the vertical position of the vehicle on a video sequence. Later, Verly and Delanoy [7] applied adaptive MM to range imagery to correct perspective distortions. Their approach consists in defining square SE such that their size depends on the distance between objects and sensor. Shih and Cheng [8] used simple and fast adaptive dilations with elliptic SE that varies its size and orientation according to local properties. A more sophisticated solution, proposed by Talbot and Appleton [9], defines pixel connectivities by complete and incomplete paths. Pinoli and Debayle [10] proposed a general adaptive neighborhood for MM: given a criterion mapping h and a tolerance $m > 0$, at each point x an adaptive neighborhood is defined containing all points y such that $|h(y) - h(x)| < m$. Lerallut et al. [11] proposed adaptive SE called amoebas. These amoebas take image gradient into account in order to adapt their shape. Morard et al. [12] proposed adaptive SE based on a region growing process. These SE have a fixed size but they adapt their shape by choosing recursively homogeneous pixels with respect to the seed pixel. Angulo [13] used the notion of counter–harmonic mean in order to propose bilateral filters which asymptotically correspond to spatially-variant morphological operators. Note that all these works are applied to MM, however they are useful to any other local operator such as convolution or non-linear filters.

In this work, a reconstruction method controlled by the evolution of a given attribute is presented (e.g. gray–level statistics, area, geodesic distances, among others). The process begins from markers, propagated over increasing quasi–flat zones, avoiding the creation of new contours on the image. Then, the propagation stops according to the attribute changes. We show that our method does not require any additional size parameter in order to determine the appropriate region, it is multi–scale and auto–dual. To demonstrate its usefulness, applications in image segmentation, adaptive SE and feature extraction are presented.

This paper is organized as follows. Section 2 presents the background on connectivity relations and quasi–flat zones. Section 3 defines propagation controlled by the evolution of attributes. Section 4 illustrates three applications. Finally, Section 5 concludes the paper.

2 Background

Connectivity relations naturally lead to partitions [14]. For example, the connectivity relation induced by the equality of gray–level divides the image into maximal connected components of constant gray–level, called flat–zones [15]. In most cases, partition in flat zones results in too many segments. A less restrictive connectivity relation can be defined adding a threshold λ. It allows to connect adjacent pixels if their gray–level difference does not exceed λ. This procedure, first introduced in image processing by Nagao et al. [16], is called quasi–flat (or λ–flat) zones labeling and it is defined as [17]:

Definition 1. *Let f be a digital gray–scale image $f : D \rightarrow V$, with $D \subset Z^2$ the image domain and $V = [0, ..., R]$ the set of gray levels. Two neighboring pixels p, q belong to the same λ–flat zone of f, if their difference $|f_p - f_q|$ is smaller than or equal to a given λ value.*

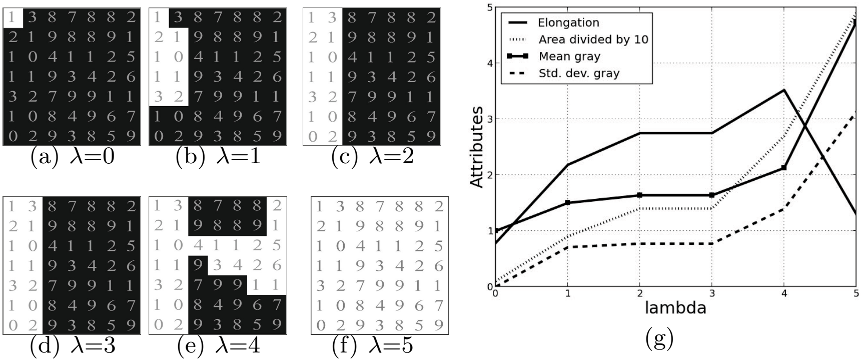
The definition of λ–flat zones is very useful in image partition, simplification and segmentation. However, it suffers from the well–known chaining effect of the single linkage clustering [18]. That is, if two distinct image objects are separated by one or more transitions going in steps having a gray–level difference lower than λ, they will be merged in the same λ–flat zone.

Several works try to restrict quasi–flat zones growth in order to prevent merging different regions. For example, Hambrusch et al. [19] proposed a technique to limit the chaining effect by introducing an additional threshold that limits gray–level variation over the whole connected component rather than just along connected paths. This relation is reflexive and symmetric, but not necessarily transitive, so it does not always lead to an image partition in the definition domain. In [20], Soille reviewed several approaches and proposed a constrained connectivity called (λ, ω, β)–connectivity. In this approach, a succession of λ–flat zones is built with increasing slope parameter λ (up to a maximum λ_{max}), none of which may have gray–level difference greater than ω and connectivity index greater than β. This method has the advantage of providing a unique partition of the image domain, which is very difficult to achieve in any other way. This method was successfully applied to hierarchical image partition and simplification. Other solutions include viscous propagations by means of geodesic reconstruction, as proposed in [21,22].

The main disadvantage of these approaches is how to tune the parameters. With the aim of simplifying this selection, we propose an attribute controlled propagation based on increasing quasi–flat zones. It consists in evaluating attribute changes during region growing in order to select the appropriate partition. For a given attribute no additional size parameter is required. In that sense, our algorithm takes advantage of a priori knowledge and intrinsic information of the image in order to define the best propagation. Let us explain it in the following section.

3 Attribute Controlled Reconstruction

The idea comes from the reconstruction of an object from a marker. Let us describe the problem with the toy example of Figure 1. Consider a marker x on the upper left corner of Figure 1(a) and its propagation by increasing λ–flat zones using 4–connected neighborhood. The propagation begins with $\lambda = 0$ and it ends when propagation reaches the whole image at $\lambda = 5$.



(a) λ=0 (b) λ=1 (c) λ=2 (d) λ=3 (e) λ=4 (f) λ=5 (g)

Fig. 1. Propagation over increasing λ–flat zones from a marker on the upper left corner

In the domain of image segmentation, the question is: *when should propagation be stopped?* Obviously, the answer depends on the specific application. Intuitively, the evolution of an attribute could be useful to make the decision.

For example, Figure 1(g) presents the evolution of four attributes: area $S(X)$, geodesic elongation[1] $E(X)$, mean gray–level $\mu_f(X)$ and standard deviation of gray–level $\sigma_f(X)$. We propose two criteria in order to stop the propagation: i) **attribute rupture**, to select the propagation such that the attribute change between two consecutive λ is maximum. ii) **maximum attribute**, to select the propagation such that the attribute is maximum.

On one hand, one can see between λ=3 and λ=4 that area increases up to 200% of its value (from 14 to 27 pixels). This great change is called an *attribute rupture*, and it can be a reason to stop the growing process. Another example occurs between λ=4 and λ=5, where ruptures are identified on $E(X)$, $\mu_f(X)$ and $\sigma_f(X)$. On the other hand, the maximum elongation occurs in λ=4. Note that for increasing attributes (e.g. area) the maximum attribute value always corresponds to the propagation on the whole image. Therefore, selecting the maximum attribute is only reasonable in the case of non-increasing attributes (e.g. geodesic elongation). For further information about increasing and non-increasing criteria, the reader is addressed to [23].

[1] For details on geodesic elongation definition, see Appendix A.

Based on Definition 1, let us introduce formal definitions for the set of increasing λ–flat zones:

Definition 2. *For all $x \in D$, let Λ_x be the set of increasing regions containing pixel x. For all $\lambda \in V$ and $j = [1, ..., n-1]$, we define $A_x(\lambda) \in \Lambda_x$ as the λ–flat zone of image f containing x:*
$A_x(\lambda) = \{x\} \cup \{q | \exists \wp = (p_1 = x, ..., p_n = q) \text{ such that } |f_{p_j} - f_{p_{j+1}}| \leq \lambda\}$

In this work λ–flat zones are arbitrarily used. However, this is not a restrictive choice since any other hierarchical partition can be used as well. Let us introduce formal definitions for attribute rupture and maximum attribute:

Definition 3. *Let $\Gamma(A_x(\lambda))$ be an attribute on the λ–flat zone of image f containing pixel x. We define λ_M and λ_R as the values for which the maximum attribute and the attribute rupture appear, respectively:*
$\lambda_M = argmax_{\lambda_i \in V} |\Gamma(A_x(\lambda_i))|$
$\lambda_R = argmax_{\lambda_i \in V} |\Gamma(A_x(\lambda_i)) - \Gamma(A_x(\lambda_{i+1}))|$

In this work, we arbitrarily analyze only one attribute at the same time. However, other statistics or combination of several attributes can be used as well. Compared with other methods, our main advantage is that no size parameter is required in order to determine the adaptive region, it is a connected operator since the λ–flat zones do not create new contours on the image [15,24], it is multi–scale, and it is auto–dual since bright, dark and intermediate gray level regions are processed at the same time.

4 Applications

To illustrate the performance of our method, three applications are presented: i) image segmentation: reconstruction by controlled propagation from markers in order to segment connected objects in range images. ii) adaptive MM: the controlled propagation is computed on a pilot image and the result is used as input–adaptive SE of each pixel on the original image, similar to [11,25]. And, iii) feature extraction: features are computed on the input–adaptive SE and they are used to characterize each point on the image with respect to its neighborhood, similar to [12].

4.1 Image Segmentation

Thanks to new 3D data availability, an increasing number of geographic applications such as Google Earth, Microsoft Virtual Earth, OpenStreetMaps and Geoportail is flourishing nowadays. Some of these applications do not only require to look realistic, but have also to be faithful to reality. Automatic urban structures segmentation is required in order to build accurate large scale 3D city models. In this section, we present an automatic facade segmentation method on 3D point clouds developed as part of TerraMobilita project (`http://www.terramobilita.fr`). It consists in four steps:

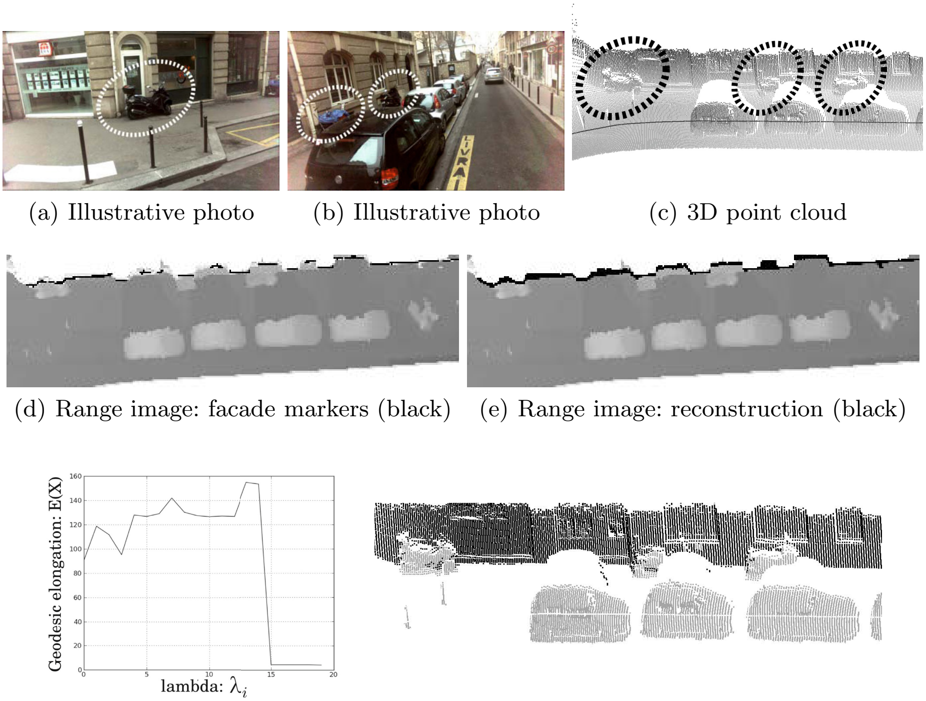
1. To project 3D data to a range image on a horizontal XY–plane (Figure 2(d)).
2. To find facade markers based on height constraints. Note that facades are the highest structures in the image.
3. To reconstruct facades from markers without including connected objects, for example motorcycles parked next to them (Figure 2(c)).
4. To reproject the result to the 3D point cloud (Figure 2(g)).

For further details about steps 1, 2 and 4, the reader is referred to [26]. Let us concentrate on the third step since the other ones are straight–forward and are out of the scope of this paper. The attribute controlled reconstruction introduced in Section 3 is used for this purpose. Facades are elongated structures in the range image. Thus reconstruction from markers stops when the geodesic elongation is maximum. Figure 2, where three motorcycles are parked next to the facade, illustrates the process. Figures 2(a) and 2(b) show pictures helpful to understand the scene. Figure 2(d) present the range image and the facade markers. Figure 2(f) shows the geodesic elongation evolution with increasing λ values. Reconstruction at λ=13 is selected, which corresponds to the maximum elongation. Note that the maximum elongation (at λ=13) and the elongation rupture (at λ=14) are almost the same connected component, thus this selection is not critical for this example. The reconstruction result is shown in Figure 2(e) on the range image and in Figure 2(g) on the 3D point cloud. One can see that the entire facade is reconstructed correctly without including connected motorcycles.

With respect to other approaches in which a parameter should be selected, our method only requires selecting an attribute, then the appropriate propagation is automatically selected. This is useful when segmenting objects with similar attributes on large databases. For example, facades are always the most elongated structures. Then, if different λ parameters are required to segment facades on different images (or even different facades on the same image), our method will adapt the parameter to the best possible value.

4.2 Adaptive Morphology

Among the different approaches in input–adaptive MM, amoebas [11] appear as a promising solution. They consist in defining a distance that depends on both the length and the gray–level differences on a neighborhood. This distance is used to define structuring elements $N(x) = \{y : d_\sigma(x, y) \leq r\}$ for each pixel on the input image. Because the amoeba distance is an increasing attribute, increasing r leads to an inclusion property useful to define pyramid of operators [27]. In fact, if the process consists in successive operators (e.g. an opening is an erosion followed by the reciprocal dilation), the SE should be the same for all of them in order to preserve mathematical properties of morphological filters, as proved by Roerdink [28]. Thus, adaptive SE are computed on a pilot image, the same for the whole process. This pilot image can be the original image or a filtered version of it since the noise can modify the SE shape.



(a) Illustrative photo (b) Illustrative photo (c) 3D point cloud

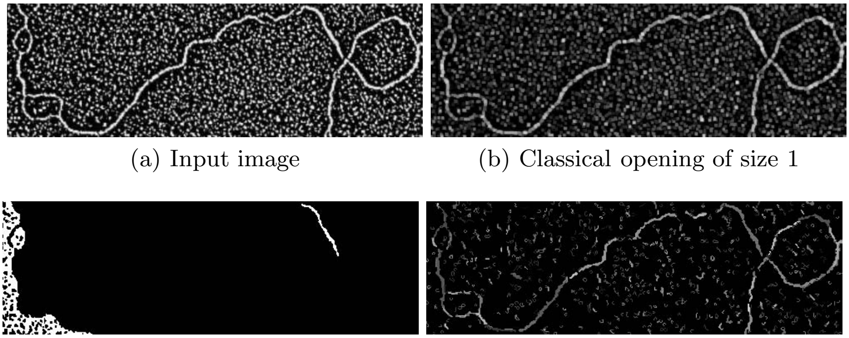
(d) Range image: facade markers (black) (e) Range image: reconstruction (black)

(f) Attribute evolution (g) Segmented 3D point cloud

Fig. 2. Segmentation of connected objects by controlled propagation from markers

In a similar way to amoebas, we apply our controlled propagation to define adaptive SE for each pixel on the pilot image. These adaptive SE are useful to filter structures according to a given attribute. For example, Figure 3 presents an opening with adaptive SE using the maximum elongation. Figure 3(c) illustrates the SE for two pixels in elongated and non-elongated regions. Figure 3(d) compares the result of an adaptive opening with respect to the classical one (Figure 3(b)). Note that elongated structures are preserved while non–elongated structures are merged with their neighborhood.

Figure 4(b) presents another example using gray–level rupture. This is useful to define SE containing pixels with similar gray–level. Figure 4(b) shows the SE for two different pixels in the image. Figure 4(c) presents the application of this adaptive SE as kernel of a non–linear filter, the median filter. Note that homogeneous regions are smoothed and high contrasted structures are preserved. Compared with amoebas and other similar works, our method does not require any additional size parameter since the SE only depends on attribute selection and the input image.



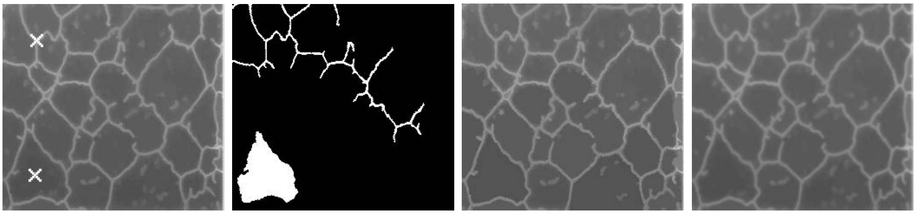
(a) Input image

(b) Classical opening of size 1

(c) Adaptive SE

(d) Adaptive opening

Fig. 3. Input–adaptive SE using the maximum elongation. In this case, the input and the pilot image are the same.



(a) Input image

(b) Adaptive SE

(c) Adaptive median

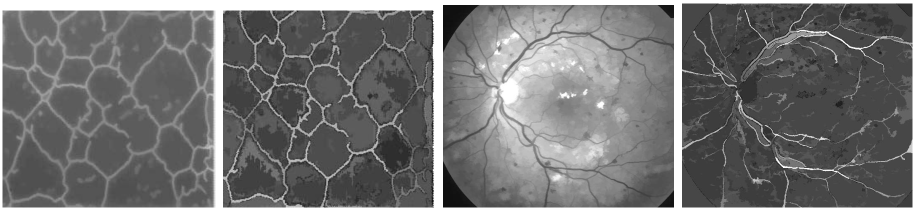
(d) Classic median

Fig. 4. Input–adaptive SE using the gray–level rupture. White cross indicates the seed pixel. In this case, the input and the pilot image are the same.

4.3 Feature Extraction

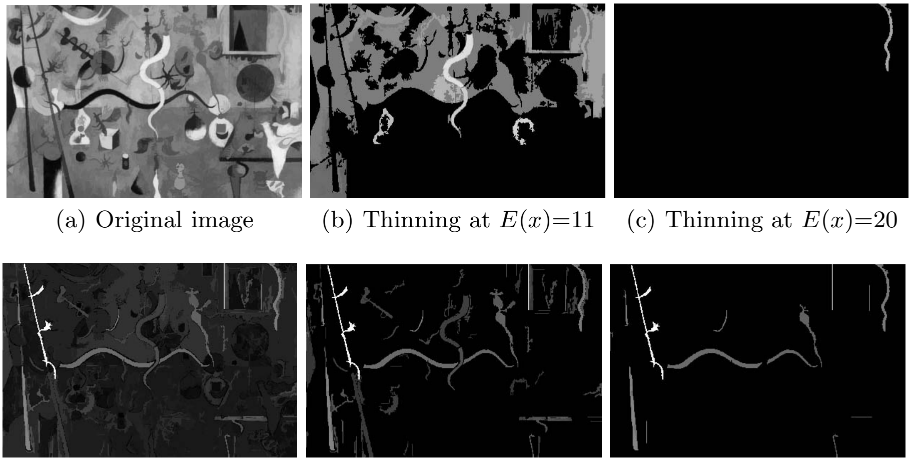
We present an application to extract features from an image based on the shape of the input–adaptive SE. To the authors knowledge, this idea was first presented by Morard et al. [12], who proposed an approach using region growing structuring elements (REGSE). For each pixel on the image, they defined a neighborhood of N pixels minimizing a homogeneity function $\rho(x)$ (e.g. gray–level difference) between adjacent pixels. REGSE can follow any homogeneous structure but cannot be multi–scale because its size has to be exactly N pixels. Finally, they used the REGSE shape to compute shape features in the image.

We propose a similar approach with our propagation method. The main comparative advantage is that parameter N is not required, because it is adaptively defined for each pixel during the propagation from it. In that sense, we use non–constant size SE that depends on the image intrinsic information. This is specially useful when the image contains objects at different scales. Additionally, remember that our propagation is a connected operator since λ–flat zones do not create new contours during propagation. This is not true for REGSE, where region growing is forced to stop at N pixels.



(a) Original image (b) Feature image (c) Original image (d) Feature image

Fig. 5. Feature images using input–adaptive SE controlled by the maximal elongation



(a) Original image (b) Thinning at $E(x)$=11 (c) Thinning at $E(x)$=20

(d) Elongation image (e) Threshold at $E(x)$=11 (f) Threshold at $E(x)$=20

Fig. 6. Extraction of elongated structures at different thresholds using geodesic thinnings [29] and thresholding on the feature image computed by our method

Consider the two examples of Figure 5. From each pixel, we compute the adaptive SE using a propagation controlled by the maximal geodesic elongation. Each pixel on the output image contains the maximal geodesic elongation of its respective adaptive SE. Note that brighter and darker structures are processed at the same time. In order to favor one of them, feature image could be weighted using gray–level input image.

Feature images are useful to assess features and segment structures by simple thresholding. Compared to geodesic thinnings [29] that uses geodesic elongation as our method does, our approach has the following advantages: i) Our feature image contains information about all objects in the scene, while geodesic thinning must be computed every time in order to extract structures at different elongations. ii) Our method, based on quasi-flat zones, deals with bright, dark or intermediate gray level regions at the same time whereas geodesic thinning, based on threshold decomposition of the image focuses only on bright objects.

Figure 6 illustrates this comparison. Figures 6(b) and 6(c) present a geodesic thinning at $E(x)$=11 and $E(x)$=20, respectively. While Figures 6(e) and 6(f) present a simple thresholding on the feature image at these same values.

5 Conclusions

We present a reconstruction controlled by the evolution of a given attribute during propagation from markers. This method is a connected operator since the propagation is done on increasing quasi–flat zones, therefore new contours are not created. Our method is auto–dual since bright, dark and intermediate gray level regions are processed at the same time. When this controlled propagation is computed from each pixel on a pilot image, input-adaptive SE can be defined for mathematical morphology operators. Additionally, features on each SE can be assessed in order to characterize the image. The main advantage of our approach is that no size parameter is required in order to determine the appropriate region.

In order to illustrate the performance of our method, three applications are presented. In the first case, controlled propagation from markers is used to separate connected objects in range images. In the second case, the propagation is computed from each pixel on a pilot image, then it is used to define input–adaptive SE that satisfies the properties of morphological filters. Moreover, if an increasing attribute is used, it satisfies the inclusion property necessary to define pyramid of operators. Finally, we take advantage of SE shape in order to assess features. In this paper we have just studied the evolution of some attributes, but the idea can be extended to other attributes and even a combination of them in a straightforward way. In our experiments, attribute selection was done according to the specific application.

By the moment, our study focus on the results rather than on the implementation performance. In our current prototype, input–adaptive SE computation is the most expensive step. However, they have to be computed only once for each image. Then, morphological operations can be computed faster.

The main drawback is the chaining effect due to transition regions. They are paths with gradual transitions connect different regions of the image in the same λ–flat zone. As consequence, the propagation can reach different objects through these paths, even for low λ values. To prevent that, it is recommended to use a filtered version of the input image as pilot image. As future work, other connected hierarchical partitions and viscous propagations will be studied.

Our propagation method is presented for gray–scales images. Future work will include its extension to color or multi-spectral images, where other metrics should be used to define propagation rules.

Acknowledgements. This work has been performed as part of TerraMobilita project.

A Appendix: Geodesic Elongation

The geodesic elongation $E(X)$ of an object X, introduced by Lantuéjoul and Maisonneuve [30], is a shape descriptor useful to characterize long and thin structures. It is defined in Equation 1, where $S(X)$ is the area and $L(X) = \sup_{x \in X}\{l_x(X)\}$ is the geodesic diameter [31], that means the longest geodesic arc of X. The longer and narrower the object, the higher the elongation. The lower bound is reached with the disk, where $E(X) = 1$. An efficient implementation can be found in [29].

$$E(X) = \frac{\pi L^2(X)}{(4S(X))} \tag{1}$$

References

1. Matheron, G.: Random Sets and Integral Geometry. John Wiley & Sons, New York (1975)
2. Serra, J.: Image Analysis and Mathematical Morphology, vol. 2. Academic Press, London (1988)
3. Maragos, P., Vachier, C.: Overview of adaptive morphology: Trends and perspectives, pp. 2241–2244 (2009)
4. Gordon, R., Rangayyan, R.M.: Feature enhancement of film mammograms using fixed and adaptive neighborhoods. Applied Optics 23(4), 560–564 (1984)
5. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)
6. Beucher, S.: Traffic Spatial Measurements Using Video Image Processing. Intelligent Robots and Computer Vision 848, 648–655 (1987)
7. Verly, J., Delanoy, R.: Adaptive mathematical morphology for range imagery. IEEE Transactions on Image Processing 2(2), 272–275 (1993)
8. Shih, F.Y., Cheng, S.: Adaptive mathematical morphology for edge linking. Information Sciences 167(1), 9–21 (2004)
9. Talbot, H., Appleton, B.: Efficient complete and incomplete path openings and closings. Image and Vision Computing 25(4), 416–425 (2007)
10. Pinoli, J.-C., Debayle, J.: General Adaptive neighborhood mathematical morphology. In: 16th IEEE International Conference on Image Processing (ICIP 2009), pp. 2249–2252 (2009)
11. Lerallut, R., Decencière, É., Meyer, F.: Image filtering using morphological amoebas. Image and Vision Computing 25(4), 395–404 (2007)
12. Morard, V., Decencière, E., Dokládal, P.: Region Growing Structuring Elements and New Operators based on their Shape. In: International conference on Signal and Image Processing (SIP 2011). ACTA Press (2011)
13. Angulo, J.: Morphological Bilateral Filtering and Spatially-Variant Adaptive Structuring Functions. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 212–223. Springer, Heidelberg (2011)
14. Serra, J.: Connectivity on Complete Lattices. Journal of Mathematical Imaging and Vision 9(3), 231–251 (1998)
15. Salembier, P., Serra, J.: Flat zones filtering, connected operators and filters by reconstruction. IEEE Transactions on Image Processing 4(8), 1153–1160 (1995)
16. Nagao, M., Matsuyama, T., Ikeda, Y.: Region extraction and shape analysis in aerial photographs. Computer Graphics and Image Processing 10(3), 195–223 (1979)

17. Meyer, F.: From connected operators to levelings. In: Heijmans, H., Roerdink, J. (eds.) Mathematical Morphology and its Applications to Image and Signal Processing. Computational Imaging and Vision, vol. 12, pp. 191–198. Kluwer Academic Publishers (1998)
18. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley Interscience (2000)
19. Hambrusch, S., He, X., Miller, R.: Parallel Algorithms for Gray-Scale Digitized Picture Component Labeling on a Mesh–Connected Computer. Journal of Parallel and Distributed Computing 20(1), 56–68 (1994)
20. Soille, P.: Constrained connectivity for hierarchical image decomposition and simplification. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(7), 1132–1145 (2008)
21. Meyer, F., Vachier, C.: Image segmentation based on viscous flooding simulation. In: Talbot, H., Beare, R. (eds.) Proceedings of the 5th International Symposium on Mathematical Morphology (ISMM 2002), Sydney, Australy, pp. 69–77 (2002)
22. Serra, J.: Viscous Lattices. Journal of Mathematical Imaging and Vision 22, 269–282 (2005)
23. Breen, E.J., Jones, R.: Attribute Openings, Thinnings, and Granulometries. Computer Vision and Image Understanding 64(3), 377–389 (1996)
24. Salembier, P., Wilkinson, M.H.F.: Connected operators. IEEE Signal Processing Magazine 26(6), 136–157 (2009)
25. Grazzini, J., Soille, P.: Adaptive Morphological Filtering Using Similarities Based on Geodesic Time. In: Coeurjolly, D., Sivignon, I., Tougne, L., Dupont, F. (eds.) DGCI 2008. LNCS, vol. 4992, pp. 519–528. Springer, Heidelberg (2008)
26. Hernández, J., Marcotegui, B.: Point Cloud Segmentation towards Urban Ground Modeling. In: Joint Urban Remote Sensing Event, Urban 2009: 5th GRSS/ISPRS, Shangai, China (2009)
27. Serra, J., Salembier, P.: Connected operators and pyramids. SPIE Image Algebra and Morphological Image Processing 2030, 65–76 (1993)
28. Roerdink, J.B.T.M.: Adaptivity and group invariance in mathematical morphology. In: Proceedings of the International Conference on Image Processing (ICIP 2009), Cairo, Egypt, pp. 2253–2256 (2009)
29. Morard, V., Decencière, E., Dokladal, P.: Geodesic attributes thinnings and thickenings. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 200–211. Springer, Heidelberg (2011)
30. Lantuéjoul, C., Maisonneuve, F.: Geodesic methods in quantitative image analysis. Pattern Recognition 17(2), 177–187 (1984)
31. Lantuéjoul, C., Beucher, S.: On the use of the geodesic metric in image analysis. Journal of Microscopy 121(1), 39–49 (1981)

On Nonlocal Mathematical Morphology

Santiago Velasco-Forero[1] and Jesús Angulo[2]

[1] ITWM - Fraunhofer Institute, Kaiserslautern, Germany
[2] CMM-Centre de Morphologie Mathématique,
Mathématiques et Systèmes, MINES ParisTech, France

Abstract. In this paper, nonlocal mathematical morphology operators are introduced as a natural extension of nonlocal-means in the max-plus algebra. Firstly, we show that nonlocal morphology is a particular case of adaptive morphology. Secondly, we present the necessary properties to have algebraic properties on the associated pair of transformations. Finally, we recommend a sparse version to introduce an efficient algorithm that computes these operators in reasonable computational time.

1 Introduction

Mathematical morphology is an approach to image analysis that characterises an image by transformations with simple geometrical interpretation [23]. The original image, denoted by **I**, is studied by its interaction with small subsets, named *structuring elements* (SEs), obtained by convolution in the max-plus algebra [9]. It has been applied successfully to a large number of fields including biomedical microscopy, material science, remote sensing, and medical imaging. The classical approach is characterised by two main properties [21]: (1) SE is fixed, i.e., does not depend on the spatial position at which it is centred; (2) the basic morphological operations are invariant under translation. This idea has been extended to grey scale images, using a complete lattice formulation [23]. This paper deals with a case of *adaptive mathematical morphology.* Adaptive mathematical morphology refers to morphological filtering techniques that adjust the SE to the local context of the image. The approach in this paper is based on the adaptive morphology framework, but where the local structuring element is "estimated" taking into consideration the whole image. We thus refer it as a "nonlocal" approach, following the terminology initiated in [1]. The term of "nonlocal morphology" has been already considered in previous works. On the one hand, Salembier in [22] proposed a straightforward generalisation of nonlocal means filter to morphological filters. As we will discuss, our starting point is similar, however the proposed non local structuring function as well as the proposed algebraic framework is totally coherent with classical morphological adjunction theory. On the other hand, Ta et al. [25] introduced a formalism of graph-based nonlocal morphology by generalising the PDE of dilation and erosion. The nonlocal PDE is solved using numerical methods which includes nonlocal distances as weights. It is obvious that such PDE-based approach does

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 219–230, 2013.

not induces a couple of adjoint dilation and erosion, and consequently their products do not involve openings and closings in the algebraic sense. We begin with a review of the extensive literature on adaptive mathematical morphology in Section 2. Section 3 covers the development of the nonlocal mathematical morphology. We introduce a simple concept of neighbour system for an image, which allow us to have algebraic morphological transformations. Section 4 introduces the idea of sparse nonlocal morphology and efficient implementations. Finally, Section 5 concludes the paper with some examples and relevant conclusions.

2 Adaptive Mathematical Morphology

Basic Definitions. Mathematical morphology considers an image $\mathbf{I}$ as a numerical function from the "spatial" space $\mathtt{E}$ to a "spectral" space $\mathbb{F}$. In the case of grey-level images, $\mathtt{E}$ is a subset of the Euclidean $\mathbb{R}^n$ or the discrete space $\mathbb{Z}^n$ (n=2 for 2D images, n=3 for 3D images), considered as the *support space* of the image, and $\mathbb{F}$ is a set of grey-levels, corresponding to the *space of values* of the image. It is assumed that $\mathbb{F} = \overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ or $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, +\infty\}$, or more specifically $\mathbb{F}$ is a closed subset of $\mathbb{R}$, for instance $\mathbb{F} = [a, b]$ for $a, b \in \mathbb{R}$. Thus, a grey-level image is represented by a function,

$$\mathbf{I} : \begin{cases} \mathtt{E} \to \mathbb{F} \\ x \mapsto \mathbf{I}(x) \end{cases} \tag{1}$$

i.e., $\mathbf{I} \in \mathrm{Fun}(\mathtt{E}, \mathbb{F})$, where $\mathrm{Fun}(\mathtt{E}, \mathbb{F})$ denotes the functions from the discrete support $\mathtt{E}$ onto the space of values of the image $\mathbb{F}$. Accordingly, $\mathbf{I}$ maps each pixel $x \in \mathtt{E}$ into a grey-level value $t \in \mathbb{F}$, $\mathbf{I}(x) = t$. Note that $\mathbb{F}$ with the natural order relation $\leq$ is a complete lattice. It is important to remark that if the $\mathbb{F}$ is a complete lattice, then $\mathrm{Fun}(\mathtt{E}, \mathbb{F})$ is a complete lattice too [24]. Morphological operators aim at extracting relevant structures of the image. This is achieved by carrying out an inquest into the image through a set of known shape called *structuring element* ($\mathtt{SE}$). The two basic words in the mathematical morphology language are *erosion* and *dilation.* They are based on the notion of infimum and supremum. For the case of flat structuring element ($\mathtt{SE}$), the flat *erosion* and *dilation* operators are defined as follow,

$$\varepsilon_{\mathtt{SE}}(\mathbf{I})(x) = \bigwedge_{y \in \mathtt{SE}(x)} \mathbf{I}(y) \quad \text{and} \quad \delta_{\mathtt{SE}}(\mathbf{I})(x) = \bigvee_{y \in \hat{\mathtt{SE}}(x)} \mathbf{I}(y), \quad x \in \mathtt{E} \tag{2}$$

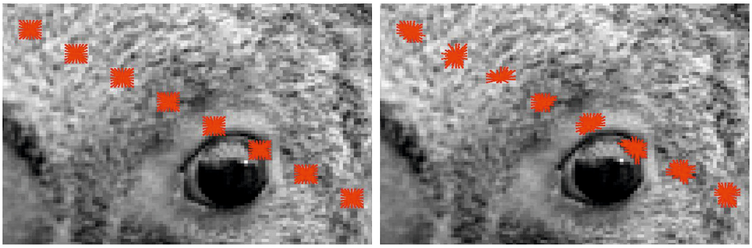
where $\mathtt{SE}(x) \subseteq \mathtt{E}$ denote the spatial neighbourhood in duced by the structuring element $\mathtt{SE}$ centred at x, and $\hat{\mathtt{SE}}$ is the transposed structuring element (i.e., reflection w.r.t. the origin).

2.1 Types of Adaptivity in Mathematical Morphology

The formulation contained in previous subsection is translation invariant in the space and in the intensity, i.e., the same processing in considered for each pixel

x in the image $\mathbf{I}$. Several ways have been analysed to define local characteristics of the image in order to locally design the SE at each point of the product space $(x \times t) \in \mathtt{E} \times \mathbb{F}$. There are different ways to define a hierarchy of approaches proposed on adaptive morphology. We use the scheme introduced by [21]. According to the adaptivity considered by the construction of the structuring element, we have two main types:

1. Location-adaptive structuring elements (variability on E [2]): The structuring element $\mathtt{SE}(x)$, depends on the location x in the image. It does not depend on the input image $\mathbf{I}(x)$. One of the earliest application that required the use of variable size SEs is the traffic control camera system [5]. This application inspired [5] to consider the perspective effect in the morphological analysis. Vehicles at the bottom of the image are closer and they appear larger than those higher in the camera. Thus, the SE should follow a law of perspective, for instance, vary linearly with its vertical position in the image. Other example is the term "locally adaptable" used in [10], for SEs as disk where the radius depend on the position of the image.
2. Input-adaptive structuring elements (variability on $\mathbb{F}$ [2]): The shape of the $\mathtt{SE}(x)$ at x depends on the local features of an image $\mathbf{I}$. We denote this kind of structuring element by $\mathtt{SE}_{\mathbf{I}}(x)$. Examples of this type of adaptive are morphological amoebas [16], intrinsic structuring elements [11], region growing structuring element [20] and morphological bilateral filtering [3]. An example of this type of adaptivity is shown in Fig.1.



(a) Classical Structuring Elements (b) Adaptive Structuring Elements

Fig. 1. SE vs $\mathtt{SE}_{\mathbf{I}}$ for some pixels

2.2 Flat Input-Adaptive Morphology

In this subsection, we limit ourselves to the case of flat input-adaptive structuring elements. Let $\mathcal{L} = \mathrm{Fun}(\mathtt{E}, \mathbb{F})$ denote the complete lattice of grey-scale functions with domain E, whose range is a complete lattice $\mathbb{F}$ of grey values. Consider the mappings $\delta : \mathcal{L} \to \mathcal{L}$ and $\varepsilon : \mathcal{L} \to \mathcal{L}$ defined by:

$$\delta_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{I})(x) := \bigvee_{y \in \mathtt{SE}_{\mathbf{I}}(x)} \mathbf{I}(y), \text{ and } \varepsilon_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{I})(x) := \bigwedge_{y \in \hat{\mathtt{SE}}_{\mathbf{I}}(x)} \mathbf{I}(y), \quad x \in \mathtt{E} \qquad (3)$$

As noted by Roerdink [21], since the neighbourhoods depend on the input $\mathbf{I}$ the mappings in (3) are in general not a dilation and erosion, i.e., they do not form an adjunction [14], hence products $\delta\varepsilon$ and $\varepsilon\delta$ are not guaranteed to satisfy the algebraic properties of opening and closing. Additionally, in [21] is given an essential conclusion: "one has to fix the adaptive neighbourhood $\mathtt{SE}_{\mathbf{I}}(x)$ once they have been derived from an initial input image $\mathbf{I}$. The one can apply the operations in (3) to any input image $\mathbf{J}$, and also use combinations of them to construct adaptive opening, closing, alternating sequentiality filters, etc." Thus, in order to have algebraic morphological operators, we need to define a set of adaptive neighbourhoods from a given image $\mathbf{I}$. That is the motivation for the first definition

Definition 1. *A structuring elements system on* $\mathbf{I} : \mathtt{E} \to \mathcal{L}$ *is a family* $\mathtt{SE}_{\mathbf{I}} = \{\mathtt{SE}_{\mathbf{I}}(x)\}_{x\in\mathtt{E}}$ *such that for all* $x, y \in \mathtt{E}$,

1. $x \in \mathtt{SE}_{\mathbf{I}}(x)$,
2. $y \in \mathtt{SE}_{\mathbf{I}}(x) \Rightarrow x \in \mathtt{SE}_{\mathbf{I}}(y)$.

The subset $\mathtt{SE}_{\mathbf{I}}(x)$ *is called the structuring element system of* x *on the image* $\mathbf{I}$.

Note that the structuring element system includes the *flat symmetric structuring elements* [23], *intrinsic structuring elements* [11], *location-adaptive structuring element* [21], and *spatially-variant morphology* [6]. A fundamental concept in mathematical morphology which plays a role of pseudo-inverse in mathematical morphology, is the adjunction [23]. The adjunction concept associated to adaptive morphology is a misleading concept. See for instance [21] for a pleasant description of this problem. The main advantage of Definition 1 is that allows to formulate the Theorem 1.

Theorem 1. *If* $\mathtt{SE}_{\mathbf{I}}$ *is a structuring element system on* $\mathbf{I}$ *then* $\delta_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{J}_1) \leq \mathbf{J}_2 \iff \mathbf{J}_1 \leq \varepsilon_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{J}_2)$, *for all* $\mathbf{I}, \mathbf{J}_1, \mathbf{J}_2 \in \mathcal{L}$

Proof. Note that the structuring element system depends only on $\mathbf{I}$. Thus, the proof is straightforward from [21].

Corollary 1. $\gamma_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{J}) := \delta_{\mathtt{SE}_{\mathbf{I}}}(\varepsilon_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{J}))$ *is an opening in the algebraic sense, i.e.* $\gamma_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{J}) \leq \mathbf{J}$ *and* $\gamma_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{J}) = \gamma_{\mathtt{SE}_{\mathbf{I}}}(\gamma_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{J}))$, *for all* $\mathbf{I}$ *and* $\mathbf{J}$ *in* $\mathcal{L}$. *Additionally, the dual operator* $\varphi_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{J}) := \varepsilon_{\mathtt{SE}_{\mathbf{I}}}(\delta_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{J}))$ *is a closing in the algebraic sense.*

Particular cases of this algebraic opening/closing definition can be found in the literature, for instance, Lerallut et al. in [16] proposed the computation of the adaptive structuring element called *amoeba* from a pilot image, which includes always the central pixel (origin). Adaptive geodesic neighbourhoods in [13] and bilateral flat structuring element [3] uses respectively a threshold over geodesic distances or convex combination of spatial distance and a pixel value distance to induce a spatial adaptive structuring element.

Remark 1. Note that $\mathtt{SE}_{\mathbf{I}}$ is fixed. That important issue, illustrated in [21], involves that if $\mathbf{J} = \delta_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{I})$, the operator $\varepsilon_{\mathtt{SE}_{\mathbf{J}}}(\mathbf{J})$ is not an opening in the algebraic sense. In our notation, that means that in general $\varepsilon_{\mathtt{SE}_{\mathbf{J}}}(\mathbf{J}) \neq \gamma_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{I})$. In practice, you cannot apply adaptive dilation followed by adaptive erosion to obtain an adaptive opening in the algebraic sense.

3 Non-flat Nonlocal Morphology

In order to fully understand how and why nonlocal morphology works, we will begin with a detailed description of nonlocal means and the theory which support the approach. Nonlocal processing refers to the general methodology of designing energies using nonlocal comparison of patches extracted in the image. Starting from the initial paper by Baudes et al. [1], nonlocal energies have proved to be efficient for many imaging problems, including denoising [1], semi-supervised classification [12] and segmentation [7]. Recently, nonlocal schemes for image processing have received a lot of attention [8]. Rather than considering only the vector associated to one pixel to compute pixel similarities, patches around these pixels are considered. These patches capture the dependencies of neighbouring pixels and thus can distinguish textural patterns. Nonlocal means filters have been proposed in [8] mainly for denoising applications. The filtering idea consists in computing a weighted average of the input image in a neighbourhood of size k:

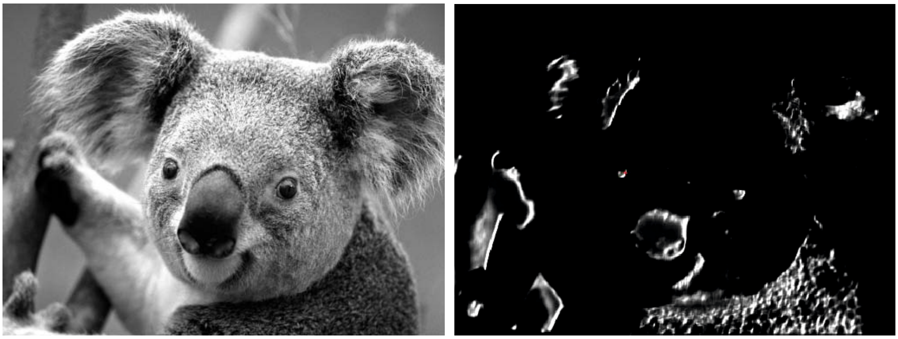
$$\begin{aligned} NLM(\mathbf{I},k)(x) &= \sum_{y\in\mathbf{I}} \mathbf{I}(y)\frac{\widetilde{\mathbf{W}_{\mathbf{I}}}(x,y)}{\sum_{z\in\mathbf{I}}\widetilde{\mathbf{W}_{\mathbf{I}}}(x,z)}, \quad x\in\mathtt{E} \\ &= \sum_{y\in\mathbf{I}} \mathbf{I}(y)\mathbf{W}_{\mathbf{I}}(x,y), \quad x\in\mathtt{E} \end{aligned} \tag{4}$$

where the weight $\mathbf{W}_{\mathbf{I}}(x,y)$ is defined by computing the similarity between a patch $\mathbf{P}$ centred around the pixel x and a patch around $y \in \mathtt{SE}_{\mathbf{I},k}(x)$

$$\widetilde{\mathbf{W}_{\mathbf{I}}}(x,y) := \widetilde{\mathbf{W}_{\mathbf{I}}}(\mathbf{P}(x),\mathbf{P}(y))) = \exp\left(-\frac{||\mathbf{P}(x)-\mathbf{P}(y)||^2}{\sigma^2}\right), \quad x\in\mathtt{E} \tag{5}$$

Here, $||\mathbf{P}||$ is the Euclidean norm of the patch $\mathbf{P}$ of size $l \times l$ as a vector in $\mathbb{R}^{l\times l}$ and σ is a smoothing parameter. Thus, pixels with similar neighbourhoods are given larger weights compared to pixels whose neighbourhoods look different. The algorithm makes explicit use of the fact that repetitive patterns appear in most of the natural images. The idea is illustrated in Fig. 2. For a review of the evolution of nonlocal modelling in imaging we recommend [15]. The "natural" morphological extension of the nonlocal means defining (4) is the version on the max-plus algebra[1], which involves replacing the convolution (i.e. $\sum_{y\in\mathbf{I}}$) by the supremum or infimum (i.e., $\bigvee_{y\in\mathbf{I}}$ or $\bigwedge_{y\in\mathbf{I}}$) and the kernel weights $\mathbf{W}_{\mathbf{I}}$ by their component-wise logarithm $\mathcal{W}_{\mathbf{I}} = \log(\mathbf{W}_{\mathbf{I}})$, i.e., :

[1] A max-plus algebra is a semiring over the union of real numbers and $-\infty$, equipped with maximum and addition as the two binary operations instead of + and × operators as in standard algebra.



(a) Original image

(b) Equation (5) for a pixel centred at the left-eye of the koala.

Fig. 2. Example of nonlocal-functional based on the grey-patch information. The centre pixel is marked by a red-cross. Original image has 384 × 512 pixels.

$$\delta_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{I})(x) = \bigvee_{y \in \mathtt{SE}_{\mathbf{I}}(x)} (\mathbf{I}(y) + \mathcal{W}_{\mathbf{I}}(x,y)), \text{ and } \varepsilon_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{I})(x) = \bigwedge_{y \in \mathtt{SE}_{\mathbf{I}}(x)} (\mathbf{I}(y) - \mathcal{W}_{\mathbf{I}}(x,y)), \tag{6}$$

for $x \in \mathtt{E}$. A similar expression to (6) was presented by Salembier in [22] without including the logarithmic transformation on $\mathbf{W}_{\mathbf{I}}$. To justify this logarithmic connection between the standard algebra $(+, \times)$ and $(\max, +)$ algebra underlying morphological operators, the reader is referred to [18,9,4]. At this point, a question arise, is the pair $(\varepsilon_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{I}), \delta_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{I}))$ an adjunction in the algebraic sense?

Definition 2. *A morphological weight system* $\mathcal{W}_{\mathbf{I}} : \mathtt{E} \times \mathtt{E} \mapsto \mathbb{R}^{+}$ *on* $\mathbf{I}$ *is a weight function such for all* $x, y \in \mathtt{E}$*,*

1. $\mathcal{W}_{\mathbf{I}}(x,x) = 0 \quad \forall x \in \mathtt{E}$*,*
2. $\mathcal{W}_{\mathbf{I}}(x,y) = \mathcal{W}_{\mathbf{I}}(y,x) \quad \forall x, y \in \mathtt{E}$*,*
3. $-\infty \leq \mathcal{W}_{\mathbf{I}}(x,y) \leq 0 \quad \forall x, y \in \mathtt{E}$*.*

In fact, we note that all conditions in Definition 2 are valid for the nonlocal weights in (4) due to the facts that $0 \leq \mathbf{W}_{\mathbf{I}} \leq 1$ and $\mathbf{W}_{\mathbf{I}}$ is a 1-diagonal and symmetric matrix.

Theorem 2. *If* $\mathtt{SE}_{\mathbf{I}}$ *is a structuring element system and* $\mathcal{W}_{\mathbf{I}}$ *a weight system on* $\mathbf{I}$ *then* $\delta_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{J}_1) \leq \mathbf{J}_2 \iff \mathbf{J}_1 \leq \varepsilon_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{J}_2)$*, for all* $\mathbf{I}, \mathbf{J}_1, \mathbf{J}_2 \in \mathcal{L}$

Proof. Firstly, note that the structuring elements system depends only on $\mathbf{I}$. The proof is straightforward from [21], however it is included to make this article globally self-contained, and then more comprehensible for the reader.

$$\begin{aligned}
&\delta_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{J}_1) \leq \mathbf{J}_2 \iff \delta_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{J}_1)(x) \leq \mathbf{J}_2(x), \forall x \in \mathtt{E} && \text{by (6)}\\
&\iff \bigvee_{y \in \mathtt{SE}_{\mathbf{I}}(x)} \mathbf{J}_1(y) + \log(\mathcal{W}_{\mathbf{I}}(x,y)) \leq \mathbf{J}_2(x), \forall x \in \mathtt{E} && \text{by max and log } -\infty \text{ and } 0.\\
&\iff \mathbf{J}_1(y) + \log(\mathcal{W}_{\mathbf{I}}(x,y)) \leq \mathbf{J}_2(x), \forall x \in \mathtt{E}, \forall y \in \mathtt{SE}_{\mathbf{I},k}(x) && \text{by 2 in Definition 2}\\
&\iff \mathbf{J}_1(y) \leq \mathbf{J}_2(x) - \log(\mathcal{W}_{\mathbf{I}}(x,y)), \forall y \in \mathtt{E}, \forall x \in \mathtt{SE}_{\mathbf{I},k}(y) && \text{by min and log}\\
&\iff \mathbf{J}_1(y) \leq \bigwedge_{x \in \mathtt{SE}_{\mathbf{I}}(y)} \mathbf{J}_2(x) - \log(\mathcal{W}_{\mathbf{I}}(x,y)), \forall y \in \mathtt{E} && \text{by 3 in Definition 2}\\
&\iff \mathbf{J}_1(y) \leq \bigwedge_{x \in \mathtt{SE}_{\mathbf{I}}(y)} \mathbf{J}_2(x) - \log(\mathcal{W}_{\mathbf{I}}(y,x)), \forall y \in \mathtt{E} && \text{by (6)}\\
&\iff \mathbf{J}_1(y) \leq \varepsilon_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{I})(y), \forall y \in \mathtt{E}\\
&\iff \mathbf{J}_1 \leq \varepsilon_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{I}), \forall y \in \mathtt{E}
\end{aligned}$$

Corollary 2. $\gamma_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{J}) := \delta_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\varepsilon_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{J}))$ *is an opening in the algebraic sense, i.e.,* $\gamma_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{J}) \leq \mathbf{J}$ *and* $\gamma_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{J}) = \gamma_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\gamma_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{J}))$*, for all* $\mathbf{I}$ *and* $\mathbf{J}$ *in* $\mathcal{L}$*. Additionally, the dual operator,* $\varphi_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{J}) := \varepsilon_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\delta_{\mathtt{SE}_{\mathbf{I}},\mathcal{W}_{\mathbf{I}}}(\mathbf{J}))$ *is a closing in the algebraic sense.*

Definition 3. *The matrix representation* $\mathcal{W}$ *of a morphological weight system* $\mathcal{W}_{\mathbf{I}}$ *given an image* $\mathbf{I}$ *with* n *pixels* $x_1, x_2, \ldots, x_n \in \mathtt{E}$ *is the square matrix of size* $n \times n$ *defined by* $\mathcal{W} = [\mathcal{W}_{\mathbf{I}}(x_i, x_j)] = [\log \mathbf{W}_{\mathbf{I}}(x_i, x_j)]$*,* $\forall i, j = 1, \ldots, n$*.*

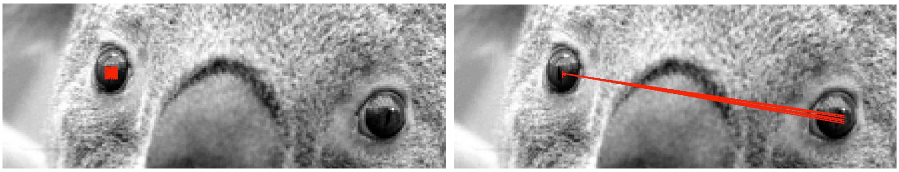
Remark 2. From Definition 2, it is easy to see that $\mathcal{W}$ in Definition 3 should be symmetric and with diagonal equal to zero. Additionally, $\mathbf{W}$ is not forced to be positive semi-definitive as in most of the linear kernel based filtering [19]. However, any positive definitive kernel induce a weight system in $\mathtt{E}$. In this section it was shown how nonlocal morphology is a particle case of adaptive morphology and a relevant conclusion was presented about this misleading term. However, the implementation of this approach requires the computation of a max-plus convolution with a full matrix $\mathcal{W}$ which in computationally intractable. We propose a solution to this bottleneck by modifying the neighbourhood connectivity mapping to connect only a small number of neighbours. Thus, it is possible to implement with almost linear complexity, as it is presented in the next section.

4 Sparse Nonlocal Morphology

In the original formulation of nonlocal morphology in [22], dilation and erosion are analysed by incorporating only the information from the k-nearest neighbours (kNNs) according to the patch distance in (5). We denote this as $\mathtt{SE}_{\mathbf{I},k}$. By simply plug-in $\mathtt{SE}_{\mathbf{I},k}$ in (6), we obtain:

$$\delta_{\mathtt{SE}_{\mathbf{I},k},\mathcal{W}_{\mathbf{I}}}(\mathbf{I})(x) = \bigvee_{y \in \mathtt{SE}_{\mathbf{I},k}(x)} (\mathbf{I}(y) + (\mathcal{W}_{\mathbf{I}}(x,y))), \quad x \in \mathtt{E} \tag{7}$$

It is important to note that the proposal in [22] fails to identify the importance issue of Properties 2 and 3 in Definition 2, i.e., the symmetry of $\mathcal{W}_{\mathbf{I}}$ and logarithmic relationship between $\mathcal{W}_{\mathbf{I}}$ and $\mathbf{W}_{\mathbf{I}}$. Note that the kNN is not a reflexive



(a) Local structuring element. (b) Sparse nonlocal structuring element.

Fig. 3. Local ($\mathtt{SE}(x)$) vs nonlocal structuring element ($\mathtt{SE}_{k,\mathbf{I}}(x)$) for the some pixels

relation, i.e., given a set of vectors $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, if $\mathbf{x}_1$ is a kNN of $\mathbf{x}_2$ on $\mathcal{X}$ does not imply that $\mathbf{x}_2$ is a k-nearest neighbours of $\mathbf{x}_1$ on $\mathcal{X}$. So, the structuring element system $\mathtt{SE}_{\mathbf{I},k}$ does not follow the Property 2 in the Definition 1. To have the symmetric property, a simple approach is to define the $\mathbf{x}_i$ as a k-NN of $\mathbf{x}_j$ based on the metric d if $d(\mathbf{x}_i, \mathbf{x}_j)$ is among the k smallest elements of the set $\{d(\mathbf{x}_i, \mathbf{x}_j) | j = 1, \ldots, i-1, i+1, \ldots, n\}$ or *viceversa*. A illustrative example of sparse structuring element is show in Fig 3. **Implementation.** A large part of the success of mathematical morphology in the imaging engineering community is due to the algorithmic developments. Very efficient algorithms have been proposed for translation invariant morphological operators for both binary and grey scale images. However, algorithms addressing the case of adaptive $\mathtt{SE}_{\mathbf{I}}$ are still very limited. Here, we proposed an efficient implementation for the case of sparse nonlocal morphology ($\mathtt{SE}_{\mathbf{I}}$) based on sparse matrices. Basically, we solve the matrix product directly in the algebra $(\max, +)$ taking advantage of the sparsity of the structuring element system. So, we define a square matrix of size $n \times n$ denoted by $\dot{\mathbf{W}}_{\mathtt{SE}_{\mathbf{I},k}}$ or by abuse of notation $\dot{\mathbf{W}} = [\mathbf{W}(i,j)]$ if $j \in \mathtt{SE}_{\mathbf{I},k}(i)$ and 0 otherwise. Thus, an adaptive dilation (erosion) may be solved efficiently as it is presented in Algorithm 1. In sparse matrices only the non-zero entries are stored. We denote as $\mathtt{findnonzero}(\mathbf{W})$ the function to obtain the non-zero entries of a sparse matrix $\mathbf{W}$. Each entry in the output represents an element $\mathbf{w}_{i,j}$ of the matrix and can be accessed by the two indices i and j. Accordingly, $\mathtt{max.row}(\mathbf{X})$ denotes the vector of the maximum of each row of $\mathbf{X}$, and $\mathtt{vec}(\mathbf{X})$ the vectorization of $\mathbf{X}$, i.e., the linear transformation to convert the matrix into a column vector.

Algorithm 1. Sparse Max-Plus Dilation

```
Require: $\mathbf{I} \in \mathbb{R}^{n_1 \times n_2}$, and a sparse square matrix $\dot{\mathbf{W}} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$.
  $\mathbf{I} = \mathtt{vec}(\mathbf{I})$                                  Vectorization of the original image.

  $[\mathbf{i}, \mathbf{j}, \mathbf{v}] = \mathtt{findnonzero}(\dot{\mathbf{W}})$              Find nonzero elements in $\dot{\mathbf{W}}$.
  for $k = 1$ to $|\mathbf{v}|$ do
    $\mathbf{O}(\mathbf{i}(k), \mathbf{j}(k)) = \mathbf{I}(\mathbf{i}(k)) + \log(\mathbf{v}(k))$       Parenthesis operation in (7).
  end for
  $\mathbf{O} = \mathtt{max.row}(\mathbf{O})$                               Maximum operation in (7).
  return $\mathbf{O}$
```

Theorem 3. *Every adaptive dilation(erosion) based on a sparse matrix $\dot{\mathbf{W}}$ can be computed in time $O(nk\log(k))$ and space $O(nk)$, where n is the number of pixels of the image and $k > 0$.*

Proof. To compute a dilation, we multiply n times the original image $\mathbf{I}$ as a vector to each of the rows of the sparse matrix $\dot{\mathbf{W}}$ of size $n \times n$ with k values different of zeros ($O(kn)$). A maximum operation should computed in rows, i.e., $O(k\log(k)n)$. Thus, the complexity of Algorithm 1 is $O(k\log(k)n)$. However, note that usually $k \ll n$ then the computation time tends to be linear $O(n)$.

Connections to Graph Theory. To warm up, let us start by recalling some graph-theoretic definitions.

- A graph $\mathcal{G}$ is pair of sets $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the elements of $\mathcal{E}$, called edges, are unordered pairs of elements from $\mathcal{V}$, called vertices.
- A sequence $x_1, x_2, \ldots, x_k$ of distinct vertices of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is called a path between x_1 and x_k if $\{x_i, x_{i+1}\} \in \mathcal{E}$ whenever $0 \leq i < k$. The length of the path is k, which is the number of edges in the path.
- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be connected if there is a path between every pair of vertices in $\mathcal{V}$.
- The adjacency matrix $\mathbf{A}_{\mathcal{G}}$ of a graph $\mathcal{G}$ with n vertices is a $n \times n$ matrix $\mathbf{A}_{\mathcal{G}} = (a_{ij})$ in which the entry $a_{ij} = 1$ if there is an edge from the vertex i to vertex j and is 0 if there is no edge from vertex i to vertex j.

The follows definitions are valid for connected graphs.

- The distance $d(x, y)$ between a pair of vertices $x, y \in \mathcal{V}$ is the length of the shortest path between these vertices.
- The eccentricity $e(x)$ of a vertex x is the maximum distance from x to any other vertex, i.e. $e(x) = \max_{y \in \mathcal{V}} d(x, y)$.
- The maximum eccentricity among all vertices of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is called the diameter, i.e. $\operatorname{diam}(\mathcal{G}) = \max_{x \in \mathcal{V}} e(x)$
- Given a set of data points $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ with $\mathbf{x}_i \in \mathbb{R}^d$. $\mathcal{G}_k(\mathbf{X}) = (\mathcal{V}, \mathcal{E})$ is a directed graph, where $\mathcal{V} = \mathbf{X}$, and the vertex $\langle \mathbf{x}_i, \mathbf{x}_j \rangle \in \mathcal{E}$ if and only if $d(\mathbf{x}_i, \mathbf{x}_j)$ is among the k smallest elements of the set $\{d(\mathbf{x}_i, \mathbf{x}_j) | j = 1, \ldots, i-1, i+1, \ldots, n\}$ or viceversa, where d is a metric.

For a disconnected graph $\mathcal{G}$, the $\operatorname{diam}(\mathcal{G})$ is defined to be the diameter of the largest connected component in $\mathcal{G}$. From a digital image $\mathbf{I}$, we define $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as an undirected graph with vertex set $\mathcal{V}$ matching the image pixels and edge set $\mathcal{E}$ consisting of unordered pairs of vertices indicating the adjacency between the image pixels according to the adaptive structuring element $\mathtt{SE}_{\mathbf{I}}$ (or $\mathtt{SE}_{\mathbf{I},k}$ for sparse nonlocal morphology). As the graph only depend on $\mathbf{I}$ and $\mathtt{SE}_{\mathbf{I}}$, we use the notation $\mathcal{G}(\mathtt{SE}_{\mathbf{I}})$ (or $\mathcal{G}(\mathtt{SE}_{\mathbf{I},k})$). Some links between the nonlocal formulation and classical graph theory are easily perceived.

- $\mathcal{G}(\mathtt{SE}_{\mathbf{I},k})$ is a $\mathcal{G}_k(\mathbf{P}_{\mathbf{I}})$, by the metric (5), where $\mathbf{P}_{\mathbf{I}}$ denotes the patch information of the image $\mathbf{I}$.

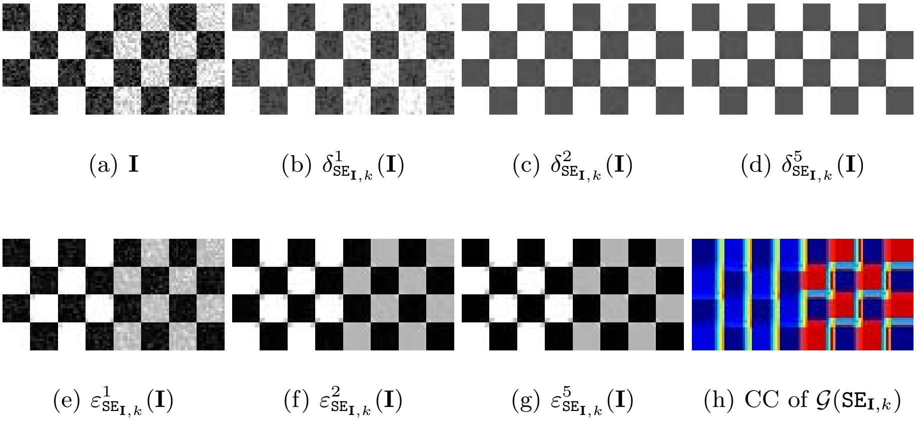
- The element-wise product between $\mathbf{A}_{\mathcal{G}(\mathtt{SE}_{\mathbf{I},k})}$ and $\mathbf{W}_{\mathbf{I}}$ is exactly the sparse matrix $\dot{\mathbf{W}}_{\mathtt{SE}_{\mathbf{I},k}}$ used in Algorithm 1.

Finally, we enunciate a less intuitive link between morphological operators and graph properties (proof is not included because of space constrains).

Theorem 4. $\delta^{i+1}_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{I}) = \delta^{i}_{\mathtt{SE}_{\mathbf{I}}}(\mathbf{I})$ *for* $i \geq$ *diam*$(\mathcal{G}(\mathtt{SE}_{\mathbf{I}}))$.

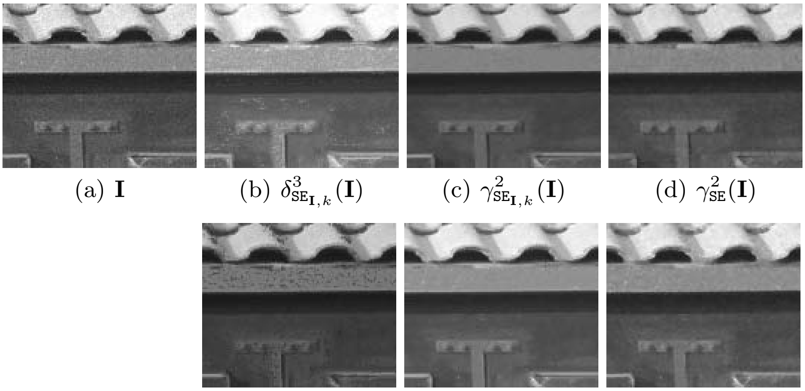
5 Experiments and Conclusions

To illustrate the effect of nonlocal morphological operators, we firstly analyse the simple geometrical case of Fig. 4. Nonlocal morphological operators perform quite well due to the connected components of $\mathcal{G}(\mathtt{SE}_{\mathbf{I},k})$, displayed in Fig.4(h), are coherent with the geometric structures of the original image. In the second example, given in Fig. 5, parameters are set to have an unique connected component. We can see that the simplification by nonlocal morphology affects only flat zones of the image, in comparison with classical morphology. However, it is important to remark that the important geometrical interpretation of the classical morphological operators is missing in the nonlocal case. Finally, visual comparison between local and nonlocal dilations and erosions can be performed in a complex image depicted in Fig. 6. To summarise, we studied a class of morphological filters which operate based on patch distance information. We also analysed in detail the requirements to have genuine adaptive morphological transformations and, as conclusion, the symmetry and logarithmic connection turns out to be the most relevant properties. Finally, we provided a fast implementation in the case of sparse nonlocal morphology which can be used in any


(a) $\mathbf{I}$ (b) $\delta^{1}_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (c) $\delta^{2}_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (d) $\delta^{5}_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$

(e) $\varepsilon^{1}_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (f) $\varepsilon^{2}_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (g) $\varepsilon^{5}_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (h) CC of $\mathcal{G}(\mathtt{SE}_{\mathbf{I},k})$

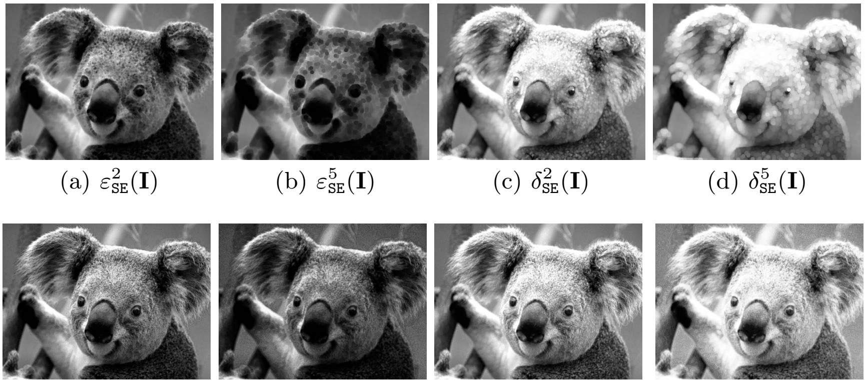
Fig. 4. Half-chessboard pattern example is a 48×96 binary image where each square has 144 pixels. The original image (a) is corrupted by impulse noise ($\sigma = .3$). Flat nonlocal morphology operators, where patches are square 3×3, and $k = 5$ are illustrated in (b)-(g). CC in (h) denotes connected components.



(a) $\mathbf{I}$ (b) $\delta^3_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (c) $\gamma^2_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (d) $\gamma^2_{\mathtt{SE}}(\mathbf{I})$

(e) $\varepsilon^3_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (f) $\varphi^2_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (g) $\varphi^2_{\mathtt{SE}}(\mathbf{I})$

Fig. 5. House pattern example is a 110 × 130 image. Patches are squares of 5 × 5, $\sigma = 200$ and $k = 5$. The k-graph contains only one connected component.



(a) $\varepsilon^2_{\mathtt{SE}}(\mathbf{I})$ (b) $\varepsilon^5_{\mathtt{SE}}(\mathbf{I})$ (c) $\delta^2_{\mathtt{SE}}(\mathbf{I})$ (d) $\delta^5_{\mathtt{SE}}(\mathbf{I})$

(e) $\varepsilon^2_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (f) $\varepsilon^5_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (g) $\delta^2_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$ (h) $\delta^5_{\mathtt{SE}_{\mathbf{I},k}}(\mathbf{I})$

Fig. 6. Examples of classical (top) and sparse nonlocal (bottom) erosion and dilations, where patches are squares 5 × 5, $\sigma = 200$ and $k = 5$. Original image in Fig. 2.

adaptive morphology. Future work includes the comparison of our approach with *nonlocal total ordering* by manifold learning introduced by [17].

References

1. Baudes, A., Coll, B., Morel, J.M.: A review of image denoising algorithms with a new one. Multiscale Modeling and Simulation 4(2), 490–530 (2005)
2. Angulo, J., Velasco-Forero, S.: Structurally adaptive math. morph. based on nonlinear scale-space decomp. Image Analysis & Stereology 30(2), 111–122 (2011)
3. Angulo, J.: Morphological bilateral filtering and spatially-variant adaptive structuring functions. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 212–223. Springer, Heidelberg (2011)

4. Angulo, J., Velasco-Forero, S.: Stochastic morphological filtering and bellman-maslov chains. In: Luengo, C., Borgefors, G. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 171–182. Springer, Heidelberg (2013)
5. Beucher, S., Blosseville, J.M., Lenoir, F.: Traffic spatial measurements using video image processing. In: Proc. Intelligent Robots and Computer Vision. SPIE (1988)
6. Bouaynaya, N., Charif-Chefchaouni, M., Schonfeld, D.: Theoretical Foundations of Spatially-Variant Mathematical Morphology Part I: Binary Images. IEEE Trans. Patt. Ana. Mach. Lear. 30(5), 823–836 (2008)
7. Bresson, X., Chan, T.F.: Non-local unsupervised variational image segmentation models. Tech. rep., UCLA CAM (2008)
8. Buades, A., Coll, B., Morel, J.M.: Image denoising methods. SIAM Review 52(1), 113–147 (2010)
9. Burgeth, B., Weickert, J.: An explanation for the logarithmic connection between linear and morph. system theory. Int. J. Comp. Vision 64(2-3), 157–169 (2005)
10. Cuisenaire, O.: Locally adaptable mathematical morphology using distance transformations. Pattern Recognition 39(3), 405–416 (2006)
11. Debayle, J., Pinoli, J.C.: Spatially adaptive morphological image filtering using intrinsic structuring elements. Image Analysis and Stereology 39(3), 145–158 (2005)
12. Gilboa, G., Osher, S.: Nonlocal Linear Image Regularization and Supervised Segmentation. Multiscale Modeling & Simulation 6(2), 595–630 (2007)
13. Grazzini, J., Soille, P.: Edge-preserving smoothing using a similarity measure in adaptive geodesic neighbourhoods. Pattern Recognition 42(10), 2306–2316 (2009)
14. Heijmans, H.: Theoretical aspects of gray-level morphology. IEEE Trans. Patt. Ana. Mach. Lear. 13(6), 568–582 (1991)
15. Katkovnik, V., Foi, A., Egiazarian, K., Astola, J.: From local kernel to nonlocal multiple-model image denoising. Inte. Journal of Comp. Vision 86, 1–32 (2010)
16. Lerallut, R., Decencière, E., Meyer, F.: Image filtering using morphological amoebas. Image and Vision Computing 4(25), 395–404 (2007)
17. Lézoray, O., Elmoataz, A.: Nonlocal and multivariate mathematical morphology. In: International Conference on Image Processing. IEEE (2012)
18. Maragos, P.: Slope transforms: theory and application to nonlinear signal processing. IEEE Transactions on Signal Processing 43(4), 864–877 (1995)
19. Milanfar, P.: Symmetrizing smoothing filters. SIAM Journal on Imaging Sciences (to appear 2013)
20. Morard, V., Decencière, E., Dokládal, P.: Region growing structuring elements and new operators based on their shape. In: Signal and Image Proc. ACTA Press (2011)
21. Roerdink, J.B.T.M.: Adaptivity and group invariance in mathematical morphology. In: ICIP, pp. 2253–2256 (2009)
22. Salembier, P.: Study on nonlocal morph. operators. In: ICIP, pp. 2269–2272 (2009)
23. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, Inc., Orlando (1982)
24. Serra, J.: Image Analysis and Mathematical Morphology vol. 2: Theoretical Advances. Academic Press (1988)
25. Ta, V., Elmoataz, A., Lezoray, O.: Nonlocal PDEs-based morphology on weighted graphs for image and data proc. IEEE Trans. Im. Proc. 20(6), 1504–1516 (2011)

Vectorial Quasi-flat Zones for Color Image Simplification

Erhan Aptoula[1], Jonathan Weber[2], and Sébastien Lefèvre[3]

[1] Okan University, Turkey
erchan.aptoula@okan.edu.tr
[2] Université de Lorraine, LORIA-UMR 7503, France
jonathan.weber@univ-lorraine.fr
[3] Université de Bretagne-Sud, IRISA, France
sebastien.lefevre@univ-ubs.fr

Abstract. Quasi-flat zones enable the computation of homogeneous image regions with respect to one or more arbitrary criteria, such as pixel intensity. They are most often employed in simplification and segmentation, while multiple strategies exist for their application to color data as well. In this paper we explore a vector ordering based alternative method for computing color quasi-flat zones, which enables the use of vectorial α and ω parameters. The interest of this vectorial strategy w.r.t marginal quasi-flat zones is illustrated both qualitatively and quantitatively by means of color simplification and segmentation experiments.

Keywords: Quasi-flat zones, Image partition, Image simplification, Color morphology, Vector orderings.

1 Introduction

For some time now quasi-flat zones [8], i.e. homogeneous image regions with respect to one or more arbitrary criteria, such as pixel intensity, have been enjoying the interest of the morphological image analysis community. Given their strong application potential, in terms of image simplification and segmentation, several definitions have been elaborated in the past with varying degrees of flexibility and efficiency, e.g. [1,4,11,14,15]; out of which the (α, ω)-zones [11] stand particularly out, due to their practical properties.

Although most of them focus on grayscale data, color extensions have been also elaborated, relying either on the marginal processing of each color channel [11], or on customized inter-pixel metrics, that take into account color specific information [15]. Inspired by the work on vectorial color mathematical morphology and the application specific advantages of vector strategies [2], we have decided to take a different direction in this context. More precisely, we focus on the (α, ω)-zones and investigate a vector-ordering based approach for them, of which the main interest w.r.t marginal quasi-flat zones, lies in a) being able to specify channel-specific local and global variation criteria and b) being able to access a much finer range of local variation criteria, while searching for the one satisfying the global criterion. The practical interest of the proposed

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 231–242, 2013.

approach is illustrated both qualitatively and quantitatively through simplification and segmentation tests respectively.

The rest of the paper is organized as follows. Section 2 provides the necessary theoretical background. We then proceed to Section 3, where we elaborate on the proposed vectorial extension to quasi-flat zones. The experiments focusing on the simplification and segmentation of color images are presented in Section 4, while Section 5 is dedicated to concluding remarks.

2 Background

2.1 Definitions

The notations presented in this section have been introduced in [9] and [10]. Let $f : E \to T$ be a digital image, where E is its definition domain, the discrete coordinate grid (usually $\mathbb{N}^2$ for a 2-D image), $\mathcal{P}(E)$ the set of all subsets of E and T the set of possible image values. In the case of a grayscale image, T can be defined on $\mathbb{R}$, but it is often defined rather on a subset of $\mathbb{Z}$, most commonly $[0, 255]$. In case of multivariate images such as color or multispectral, T is defined on $\mathbb{R}^n$ or $\mathbb{Z}^n$, where $n > 1$ denotes the number of image channels. We denote $f(p)$ the intensity of pixel p in grayscale images and $f^b(p)$ the intensity in the band b of pixel p in multivariate images.

Definition 1. *A partition $\boldsymbol{P}$ of E is a mapping $p \to \boldsymbol{P}(p)$ from E into $\mathcal{P}(E)$ such that:*

1. *$\forall p \in E,\ p \in \boldsymbol{P}(p)$;*
2. *$\forall p, q \in E,\ \boldsymbol{P}(p) = \boldsymbol{P}(q)$ or $\boldsymbol{P}(p) \cap \boldsymbol{P}(q) = \emptyset$.*

The above term $\boldsymbol{P}(p)$ indicates the part of $\boldsymbol{P}$ which contains the pixel p. We note that:

$$\bigcup_{p \in E} \boldsymbol{P}(p) = E \tag{2.1}$$

The partition of an image can be obtained through different methods. Here we consider Quasi-Flat Zones which rely on the concept of connectivity. We define a connection as the family of all the sets of a space that are connected according to some connectivity criterion.

Definition 2. *A connection $\mathcal{C}$ is any family in $\mathcal{P}(E)$ such that:*

1. *$\emptyset \in \mathcal{C}$;*
2. *$\forall p \in E, \{p\} \in \mathcal{C}$;*
3. *for each family $\{\mathcal{C}_i, i \in L\} \subseteq \mathcal{C}, \bigcap_{i \in L} \mathcal{C}_i \neq \emptyset,\ \bigcup_{i \in L} \mathcal{C}_i \in \mathcal{C}$, where L is an index set.*

Partitioning an image using a connection $\mathcal{C}$ is achieved through a search of the connected components which are of maximal extent according to the connection $\mathcal{C}$. A connected component $C \subseteq E$ is of maximal extent if there is no other set $C' \supset C$ such that $C' \subseteq E$ and $C' \in \mathcal{C}$.

In practice, identifying connected components of maximal extent requires to define a dissimilarity measure between two pixels. Let us assume that each pixel $q \in E$ can be described by some attribute written $A(q)$ (e.g. its intensity) and consider a given r-norm $\|\cdot\|_r$. A dissimilarity can then be measured between a couple of pixels following:

$$q, q' \in E, d(q, q') = \|A(q) - A(q')\|_r \tag{2.2}$$

Two adjacent pixels p and q will then belong to the same connected component C if $d(p, q) < \mathcal{S}$, $\mathcal{S}$ being a dissimilarity threshold. If they are not adjacent, the dissimilarity is measured along a path linking them. A path $\pi(p \rightsquigarrow q)$ of length N_π between any two elements $p, q \in E$ is a chain (noted as $\langle \ldots \rangle$) of pairwise adjacent pixels:

$$\pi(p \rightsquigarrow q) \equiv \langle p = p_1, p_2, \ldots, p_{N_\pi - 1}, p_{N_\pi} = q \rangle, \tag{2.3}$$

Definition 3. *Let $\Pi \neq \emptyset$ be the set of all possible paths between a pair of pixels p and q. The minimum dissimilarity metric with respect to some pre-specified pixel attribute, is the ultrametric functional given by:*

$$\widehat{d}(p, q) = \bigwedge_{\pi \in \Pi} \left\{ \bigvee_{i \in [1, \ldots, N_\pi - 1]} \left\{ d(p_i, p_{i+1}) \mid \langle p_i, p_{i+1} \rangle \, \mathit{subchain\ of} \, \pi(p \rightsquigarrow q) \right\} \right\} \tag{2.4}$$

In other words, the dissimilarity measured between two pixels p to q is the lowest cost of a path from p to q, with the cost of a path being defined as the maximal dissimilarity between pairwise adjacent pixels along the path.

2.2 Quasi-flat Zones

Since several quasi-flat zone definitions have been proposed in the literature, we consider here only the two main definitions, namely the α-zones C^α and the (α, ω)-zones $C^{\alpha,\omega}$. The interested reader is referred to Ref. [11] for more details. Quasi-flat zones are defined by extension of flat zones (generally called connected components), which are defined as:

$$C(p) = \{p\} \cup \{q | \widehat{d}(p, q) = 0\} \tag{2.5}$$

The flat zones cluster connected sets of adjacent pixels with same attribute values (generally the intensity). This very restrictive definition leads to very small sets of pixels when dealing with natural images. So, a local range parameter (α) has been introduced to tolerate a dissimilarity between adjacent pixels in order to obtain wider connected components, thus leading to the definition of α-zones:

$$C^\alpha(p) = \{p\} \cup \{q | \widehat{d}(p, q) \leq \alpha\} \tag{2.6}$$

The C^α of a pixel p is then the set of pixels to which p is linked through at least one path where the dissimilarity between adjacent pixels is less or equal to α. Note that flat zones are a particular case of C^α where $\alpha = 0$. Let us observe that segmenting an image into C^α with $\alpha > 0$ may result in an undersegmentation phenomenon. Even with small α values, it may lead to the so-called "chaining effect" (see [11]).

In order to counter this problem, several new quasi-flat zone definitions based on C^{α} have been elaborated. We focus here on $C^{\alpha,\omega}$ [11] which relies on a global range ω and the following hierarchical property of C^{α}:

$$\forall \alpha' \leq \alpha, C^{\alpha'}(p) \subseteq C^{\alpha}(p) \tag{2.7}$$

It leads to the following definition of $C^{\alpha,\omega}$:

$$C^{\alpha,\omega}(p) = \max\{C^{\alpha'}(p) \mid \alpha' \leq \alpha \text{ and } R(C^{\alpha'}(p)) \leq \omega\} \tag{2.8}$$

where $R(C^{\alpha})$ is the maximal difference between the attributes of two pixels of C^{α}. So, the $C^{\alpha,\omega}$ of a pixel p is the widest $C^{\alpha'}$ (i.e. built with the highest $\alpha' \leq \alpha$ thanks to property 2.7) where the maximal inter-pixel difference is less than or equal to ω.

3 Color Quasi-flat Zones

The extension of quasi-flat zones to multivariate and more specifically to color images although not straightforward is not as difficult as color mathematical morphology [2]. Multiple approaches have been elaborated in the past, with various advantages and disadvantages, that we are going to recall shortly in this section, before presenting our method.

3.1 Related Work

Having limited our scope to α- and (α, ω)-zones, the approaches that have been conceived so far for their extension to color images, fall into two major categories denoted as marginal and vectorial [2]. Given a color image, its marginal quasi-flat zone processing, leads to channel-wise computations where each dimension of color pixels is handled independently. In practice this is equivalent to computing the partition of each of the three dimensions separately, which most often will lead to three incoherent partitions that will need to be subsequently merged w.r.t. some arbitrary criterion.

For instance the color extension of C^{α} proposed by Angulo and Serra [1] is based on a polar color space, where colors are represented in terms of hue, saturation and luminance, thus providing an effective distinction of chromatic (i.e. hue and saturation) from achromatic information (i.e. luminance). The challenge in this context consists in combining these two types of information, by employing hue only for "sufficiently" saturated colors, since it is undefined for zero saturation. Thus, the merging step of marginal quasi-flat zones lends itself perfectly well for this task, since by computing the zones on luminance and hue channels only, all that remains is to realize the merging of the resulting two partitions, by employing the original image's saturation levels. Specifically they have thresholded the saturation, hence obtaining a binary saturation map, denoting the areas of high saturation where hue based quasi-flat zones are to be used, and areas of low saturation where luminance based quasi-flat zones are preferred.

Another partition merging method has been presented by Weber [12], who employs a voting mechanism, adaptable to both C^{α} and $C^{\alpha,\omega}$. Specifically, given n marginally computed partitions, two pixels are then considered in the same quasi-flat zone, if and only if they belong to the same quasi-flat zone in at least ι channels out of n. The ι parameter in this case constitutes a way of controlling the level of oversegmentation.

Soille [11] on the other hand, has chosen to conduct the merging procedure implicitly during the quasi-flat zone computation stage. More precisely, the color pixel attribute difference is once again calculated channel-wise, using marginal vector differences. This way, the difference of two color pixels (or attributes thereof) is considered as less or greater than $\boldsymbol{\alpha} \in \mathbb{R}^n$, if and only if the same is valid across all image channels:

$$\forall i \in [1, n],\ \forall p, q \in E,\ |f^i(p) - f^i(q)| \leq \alpha_i \Leftrightarrow d(p, q) \leq \boldsymbol{\alpha} \tag{3.1}$$

The same principle can be employed when dealing with $C^{\alpha,\omega}$ in order to carry out the additional comparison $R(C^{\alpha}(p)) \leq \boldsymbol{\omega}$. In other words, a color pixel attribute difference is considered less or equal than a vectorial $\boldsymbol{\omega} \in \mathbb{R}^n$, if and only if its marginal components are each and every one less or equal than the respective marginal components of $\boldsymbol{\omega}$. However, establishing the multivariate version of $C^{\alpha,\omega}$, requires additionally a vector comparison scheme for conducting the comparison $\boldsymbol{\alpha}' \leq \boldsymbol{\alpha}$ of Eq. (2.8). And at this point, the lack of totality renders marginal ordering insufficient for this task, since there can be vectors that are incomparable.

To overcome this, Soille [11] has suggested the use of only $\boldsymbol{\alpha} = \alpha\mathbf{1}$ vectors, thus artificially inducing a total ordering among the $\boldsymbol{\alpha}$ values: $[0, 0, 0]^T \leq [1, 1, 1]^T \leq \ldots$. Consequently, this way one effectively filters out all $\boldsymbol{\alpha}$ vectors that are marginally incomparable, thus obtaining a valid multivariate $C^{\alpha,\omega}$ definition, denoted as $C^{\alpha,\omega}_{Soille}$, albeit with a limited value domain for $\boldsymbol{\alpha}$.

Leaving marginal approaches aside, the vectorial strategy manipulates each color pixel as a whole, thus taking into account inter-channel relations and fully avoiding the merging step. This approach has been adopted notably by Zanoguera [15], who computes color pixel attribute differences by means of various norms in different color spaces such as RGB, CIELAB, YUV and HSV, thus employing the scalar order whenever a comparison is required between the scalar α and ω and the also scalar color pixel attribute differences. This approach provides a means of controlling the contribution of each channel into the overall quasi-flat zone computation, by means of the distance measure under consideration.

However, her definition becomes quite impractical when dealing with $C^{\alpha,\omega}$. To explain, in order to implement the global variation criterion ω, according to the definition of Eq. (2.8) one needs to be able to compare it against the maximal vector pixel attribute difference $R(C^{\alpha})$. Although these attribute vector differences can be of course computed through the chosen color distance measure, determining their maximal value on the other hand, constitutes a serious efficiency issue, since the attribute distances of all possible vector pixel couples are required for every pixel added to a quasi-flat zone; hence leading to an eventually prohibitive computational cost.

3.2 A Purely Vectorial Approach

To address the problem brought by the definition of quasi-flat zones on color images, existing approaches either proceed with a marginal approach ignoring inter-channel correlation and demanding a merging of partitions, or consist of a vectorial approach using color distances that still however employ scalar α and ω values. Consequently, if one decides to use channel specific arguments, then an alternative strategy becomes necessary, where both arguments $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$ are vectorial. Before we proceed into the details of this approach, let us elaborate on our motivation.

Although using a different α and/or ω value per channel provides clearly a higher level of customization, which is at this point unclear how to determine optimally, in our opinion the main advantage of using vectorial arguments, is that it provides a finer search space for $C^{\alpha,\omega}$ when trying to determine the greatest α vector satisfying the global variation criterion ω, thus potentially leading to zones of higher quality. This claim will be put to test in the following section.

In order to be able to use arbitrary and vectorial $\boldsymbol{\alpha}$ along with C^{α} there is one fundamental issue to resolve, and that is the modification of the pixel attribute difference d of Eq. (2.2) into $d_{\preceq}$, so as to render it capable of producing a result comparable w.r.t. an arbitrary ordering $\preceq$ against $\boldsymbol{\alpha}$. To explain, d has to now accommodate vectorial pixels. And their resulting dissimilarity needs to be computed in such a way that it will become possible to compare it against an also vectorial $\boldsymbol{\alpha}$ using the ordering $\preceq$.

This can be resolved in at least two ways, depending whether the pixel attribute difference output is scalar or vectorial. If scalar, we can use the same tactic as Soille [11]; to explain, after mapping a given couple of pixels p and q into an arbitrary scalar attribute difference $d(p,q) \in \mathbb{R}$, we can conduct the local variation criterion control against $\boldsymbol{\alpha}$ as: $d(p,q) \cdot \mathbf{1} \preceq \boldsymbol{\alpha}$.

Alternatively, if we desire the pixel attribute difference to remain vectorial, we can resort to comparing the ranks of the vectors under consideration w.r.t. the ordering under consideration. Given an arbitrary vector ordering $\preceq$ imposed on a multi-dimensional space T, we denote by $\text{rank}_{\preceq} : T \rightarrow \mathbb{N}$ the function associating each vector with its position in that space w.r.t. $\preceq$. The smallest vector in T w.r.t. $\preceq$ will have a rank 0, the next a rank of one, etc. In particular, all the vectors present in the multivariate set of pixel values are projected onto a space-filling curve, a curve passing from all the points of the multi-dimensional discrete space, where each vector has its own unique rank. Of course this requires a total ordering. Hence the attribute difference of vector pixels to be compared against $\boldsymbol{\alpha}$ using a custom $\preceq$ ordering becomes:

$$\forall p, q \in E,\ d_{\preceq}(p,q) = |\text{rank}_{\preceq}(\boldsymbol{f}(p)) - \text{rank}_{\preceq}(\boldsymbol{f}(q))| \tag{3.2}$$

Consequently the new definition of C^{α} becomes:

$$C^{\boldsymbol{\alpha}}_{\preceq}(p) = \{p\} \cup \{q \mid \widehat{\mathbf{d}_{\preceq}}(p,q) \leq \text{rank}_{\preceq}(\boldsymbol{\alpha})\} \tag{3.3}$$

where

$$\widehat{\mathbf{d}}(p,q) = \bigwedge_{\pi \in \prod} \left\{ \bigvee_{i \in [0,\ldots,N_{\pi}-1]} \{d_{\preceq}(p_i, p_{i+1}) | \langle p_i, p_{i+1} \rangle \text{ subchain of } \pi(p \rightsquigarrow q)\} \right\} \tag{3.4}$$

The adaptation of $C^{\alpha,\omega}$ to vectorial $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$ is identical. Expression $R_{\preceq}(C^{\boldsymbol{\alpha}}(p))$ now computes the greatest rank difference w.r.t. $\preceq$ among all the vectorial pixels present within $C^{\boldsymbol{\alpha}}(p)$, and the resulting rank is compared against that of $\boldsymbol{\omega}$ in order to determine if the quasi-flat zone violates or not the global variation criterion.

Moreover, each time a pixel p of a quasi-flat zone $C^{\boldsymbol{\alpha}}$ is tested against the global criterion $\boldsymbol{\omega}$ and the test fails, one needs to fall back to the immediately next smaller $\boldsymbol{\alpha}' \preceq \boldsymbol{\alpha}$. Yet, this cannot happen unless the set of $\boldsymbol{\alpha}$ vectors is totally ordered. In conclusion, the multivariate version of $C^{\alpha,\omega}$ based on vectorial arguments, requires a total ordering to be imposed on the set of $\boldsymbol{\alpha}$ vectors:

$$C_{\preceq}^{\boldsymbol{\alpha},\boldsymbol{\omega}}(p) = \max\{C_{\preceq}^{\boldsymbol{\alpha}'}(p) \mid \boldsymbol{\alpha}' \preceq \boldsymbol{\alpha} \text{ and } R_{\preceq}(C^{\boldsymbol{\alpha}'}(p)) \leq \text{rank}_{\preceq}(\boldsymbol{\omega})\} \tag{3.5}$$

which can be considered as the cost of the flexibility for employing channel specific global and local variation criteria.

In conclusion, total vector orderings effectively enable the computation of color quasi-flat zones using vector parameters. Although this equips the tool under consideration with great potential in terms of customization (e.g. inter-channel relation modeling, channel-specific parameters, etc.), without sacrificing theoretical validity, it also increases the burden of configuration, as setting these arguments optimally constitutes undoubtedly a challenge.

4 Experiments

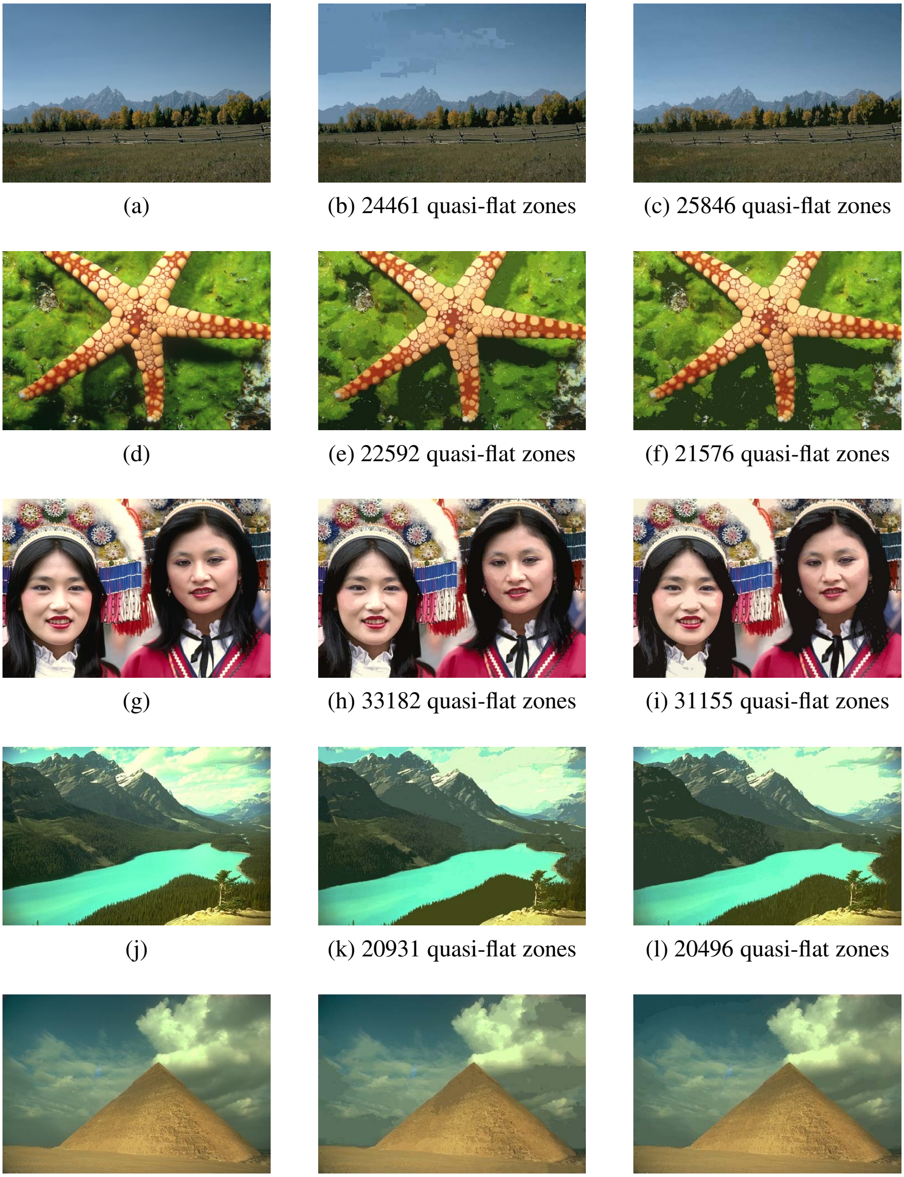
In this section, we present the results of simplification and segmentation experiments that have been conducted in order to compare the performance of the proposed vectorial color quasi-flat zone extension against the marginal strategy. For the sake of simplicity, we consider the RGB color space. The main challenge of providing a total order for the set of $\boldsymbol{\alpha}$ vectors may be resolved by means of the Euclidean norm. However, a pitfall with norm based orderings, is their lack of anti-symmetry, which leads to pre-orderings. A way around this problem can be to supplement the Euclidean norm with a lexicographical comparison $(\leq_L)$ [3], as follows:

$$\forall\, \mathbf{v}, \mathbf{v}' \in \mathbb{R}^3, \;\; \mathbf{v} \preceq_{rgb} \mathbf{v}' \Leftrightarrow [\|\mathbf{v}\|, v_1, v_2, v_3]^T \leq_L [\|\mathbf{v}'\|, v_1', v_2', v_3']^T \tag{4.1}$$

thus rendering $\preceq_{rgb}$ a total ordering, and enabling the computation of the widest quasi-flat zone required for defining $C_{\preceq_{rgb}}^{\boldsymbol{\alpha},\boldsymbol{\omega}}$, while using $d_{\preceq_{rgb}}$. This ordering can be of course used equally well during the calculation of the minimal dissimilarity metric, as well as for the comparison against the global variation criterion $\boldsymbol{\omega}$, thus effectively leading to a multivariate solution specifically adapted for the RGB color space.

4.1 Simplification

In terms of qualitative comparison we have conducted a series of image simplification tests employing images from the Berkeley Segmentation Dataset (BSD) [7]. In particular, we compare the proposed vectorial approach against $C_{Soille}^{\alpha,\omega}$, using five images of the said dataset. Image simplification is realized by producing quasi-flat zones on the images, while each quasi-flat zone is represented by the mean value of its pixels. As to the α and ω parameters of both tools, they are distinct and have been arranged so as to obtain similar numbers of quasi-flat zones. The initial results are shown in Fig. 1.



(a)

(b) 24461 quasi-flat zones

(c) 25846 quasi-flat zones

(d)

(e) 22592 quasi-flat zones

(f) 21576 quasi-flat zones

(g)

(h) 33182 quasi-flat zones

(i) 31155 quasi-flat zones

(j)

(k) 20931 quasi-flat zones

(l) 20496 quasi-flat zones

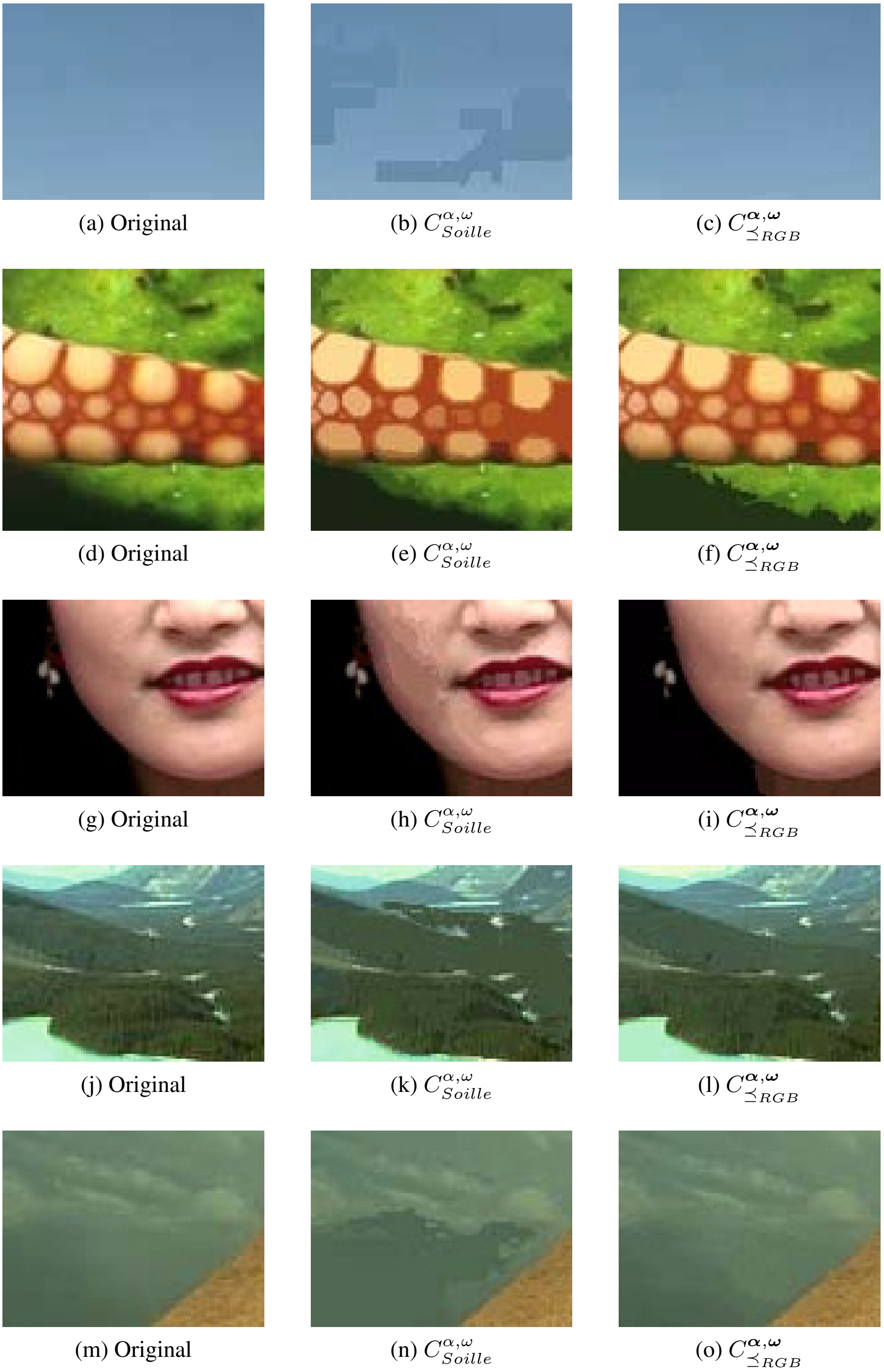
(m)

(n) 29766 quasi-flat zones

(o) 28562 quasi-flat zones

Fig. 1. Comparison of approaches in terms of image simplification: (left column) original images, (middle column) results of $C^{\alpha,\omega}_{Soille}$ in RGB, (right column) results of $C^{\boldsymbol{\alpha},\boldsymbol{\omega}}_{\preceq_{rgb}}$ in RGB

Since one can hardly distinguish the differences among the results of Fig. 1, with the exception perhaps of the relatively large artifact in Fig. 1b, we additionally provide zoomed versions of the images in Fig. 2. Judging from the obtained results, one can confirm that in these cases the proposed approach seems to achieve superior quality



(a) Original (b) $C^{\alpha,\omega}_{Soille}$ (c) $C^{\boldsymbol{\alpha},\boldsymbol{\omega}}_{\preceq_{RGB}}$

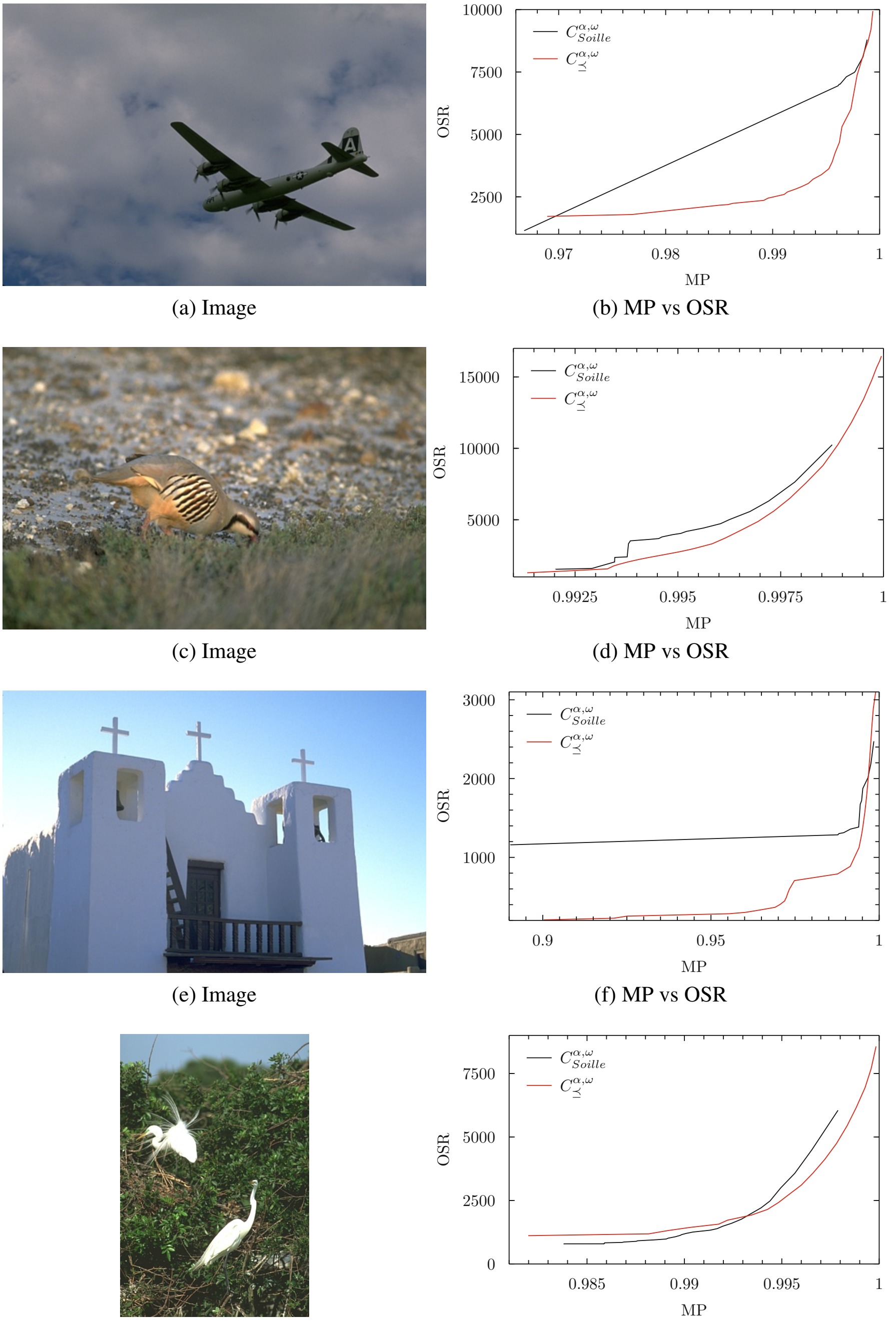
(d) Original (e) $C^{\alpha,\omega}_{Soille}$ (f) $C^{\boldsymbol{\alpha},\boldsymbol{\omega}}_{\preceq_{RGB}}$

(g) Original (h) $C^{\alpha,\omega}_{Soille}$ (i) $C^{\boldsymbol{\alpha},\boldsymbol{\omega}}_{\preceq_{RGB}}$

(j) Original (k) $C^{\alpha,\omega}_{Soille}$ (l) $C^{\boldsymbol{\alpha},\boldsymbol{\omega}}_{\preceq_{RGB}}$

(m) Original (n) $C^{\alpha,\omega}_{Soille}$ (o) $C^{\boldsymbol{\alpha},\boldsymbol{\omega}}_{\preceq_{RGB}}$

Fig. 2. Details of image simplification



(a) Image

(b) MP vs OSR

(c) Image

(d) MP vs OSR

(e) Image

(f) MP vs OSR

(g) Image

(h) MP vs OSR

Fig. 3. OSR and MP based segmentation comparison for marginal ($C^{\alpha,\omega}_{Soille}$) and vectorial ($C^{\boldsymbol{\alpha},\boldsymbol{\omega}}_{\preceq_{rgb}}$) quasi-flat zones using four images of the Berkeley Segmentation Dataset

level, as it leads to less simplification artifacts (Figs. 2b and 2c), good preservation of details (Figs. 2e and 2f where the blobs of the starfish arms are better preserved) and smoother variations (Figs. 2h and 2i where color variation on the cheek of the girl is smoother) w.r.t. $C^{\alpha,\omega}_{Soille}$; moreover, all are achieved with similar flat zone numbers. However, we can also observe that dark-green transition in the background region of Figs. 2e and 2f and the transition between the background and the neck of the girl in Figs. 2h and 2i are smoother with Soille's approach while still presenting visible variation levels.

4.2 Segmentation

We also test our approach in the segmentation context, where the resulting partitions are evaluated by means of two criteria: the oversegmentation ratio (OSR) [5] and maximal precision (MP) [6]. OSR measures directly the degree of over-segmentation. For MP on the other hand, each quasi-flat zone is associated with the reference region with which it shares the highest number of pixels. We then measure a pixel-based precision by computing the ratio of well-segmented pixels. Hence, by using both MP and OSR, an effective evaluation and comparison can be achieved, with the ultimate goal being both the minimization of OSR and maximization of MP. These metrics are particulary adapted to the evaluation of quasi-flat zones as a preliminary segmentation step [13].

To this end, we have selected four images of the BSD, using which we computed the plots of OSR against MP, shown in Fig. 3. These preliminary results indicate that our approach is superior to the marginal strategy as it leads to lower over-segmentation and higher precision levels. However, its performance does not appear to be consistent across all images, but rather image dependent; a fact which urges us to conduct further investigation on this matter. As a matter of fact, this comes as no surprise, as it confirms the results at [2], where the performance of vectorial operators had been also established to be image-dependent in the content-description context.

5 Conclusion

The strong application potential of quasi-flat zones for image simplification and superpixel creation, has been the main motivation behind the high volume of work concentrating on them lately. In this paper we focus on a vectorial strategy for their application to color images, in an effort to enable the use of channel-specific local and global variation criteria. The proposed approach, which relies on a vector ordering scheme, has been tested both qualitatively and quantitatively in the contexts of image simplification and segmentation, against the marginal strategy.

Although both simplification and segmentation tests indicate our approach to be superior, its performance is not consistent across all tested images. On one hand, its superiority confirms our original motivation for using vectorial arguments, as it enables exploiting a finer search space for finding the optimal $\boldsymbol{\alpha}$ vector satisfying the global variation criterion. On the other hand, the image-specific behavior of color orderings, as observed in different contexts [2], renders this approach relatively impractical for general use, since the optimal choice of vector orderings as well as the setting of vector parameters can be challenging.

Consequently, future work will focus principally on how to optimally determine the color ordering as well as the vector parameters under consideration. Moreover, we also plan to conduct a more rigorous experimentation in order to detect the limits of our approach.

References

1. Angulo, J., Serra, J.: Color segmentation by ordered mergings. In: Proceedings of the IEEE International Conference on Image Processing, Barcelona, Spain, pp. 125–128 (September 2003)
2. Aptoula, E., Lefèvre, S.: A comparative study on multivariate mathematical morphology. Pattern Recognition 40(11), 2914–2929 (2007)
3. Aptoula, E., Lefèvre, S.: Alpha-trimmed lexicographical extrema for pseudo morphological image analysis. Journal of Visual Communication and Image Representation 19(3), 165–174 (2008)
4. Brunner, D., Soille, P.: Iterative area filtering of multichannel images. Image and Vision Computing 25(8), 1352–1364 (2007)
5. Carleer, A., Debeir, O., Wolff, E.: Assessment of very high spatial resolution satellite image segmentations. Photogrammetric Engineering & Remote Sensing 71(11), 1285–1294 (2005)
6. Derivaux, S., Forestier, G., Wemmert, C., Lefèvre, S.: Supervised image segmentation using watershed transform, fuzzy classification and evolutionary computation. Pattern Recognition Letters 31(15), 2364–2374 (2010)
7. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada, vol. 2, pp. 416–425 (July 2001)
8. Meyer, F.: The levelings. In: Proceedings of the International Symposium on Mathematical Morphology, Amsterdam, The Netherlands, pp. 199–206 (June 1998)
9. Ouzounis, G.K., 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)
10. Serra, J. (ed.): Image Analysis and Mathematical Morphology, vol. 2: Theoretical Advances, 1st edn. Academic Press (February 1988)
11. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Trans. on Pattern Analysis and Machine Intelligence 30(7), 1132–1145 (2008)
12. Weber, J.: Segmentation morphologique interactive pour la fouille de séquences vidéo. Ph.D. thesis, Université de Strasbourg, France (2011)
13. Weber, J., Lefèvre, S.: Fast quasi-flat zones filtering using area threshold and region merging. Journal of Visual Communication and Image Representation 24(3), 397–409 (2013)
14. Weber, J., Lefèvre, S., Gançarski, P.: Interactive video segmentation based on quasi-flat zones. In: International Symposium on Image and Signal Processing and Analysis (ISPA), Dubrovnik, Croatia, pp. 265–270 (September 2011)
15. 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)

Morphology for Color Images via Loewner Order for Matrix Fields

Bernhard Burgeth[1] and Andreas Kleefeld[2]

[1] Saarland University,
Faculty of Mathematics and Computer Science
66041 Saarbrücken, Germany
burgeth@math.uni-sb.de

[2] Brandenburg Technical University Cottbus,
Faculty of Mathematics, Natural Sciences and Computer Science
03046 Cottbus, Germany
kleefeld@tu-cottbus.de

Abstract. Mathematical morphology is a very successful branch of image processing with a history of more than four decades. Its fundamental operations are dilation and erosion, which are based on the notion of a maximum and a minimum with respect to an order. Many operators constructed from dilation and erosion are available for grey value images, and recently useful analogs of these processes for matrix-valued images have been introduced by taking advantage of the so-called Loewner order. There has been a number of approaches to morphology for vector-valued images, that is, colour images based on various orders, however, each with its merits and shortcomings. In this article we propose an approach to (elementary) morphology for colour images that relies on the existing order based morphology for matrix fields of symmetric 2×2-matrices. An RGB-image is embedded into a field of those 2×2-matrices by exploiting the geometrical properties of the order cone associated with the Loewner order. To this end a modification of the HSL-colour model and a relativistic addition of matrices is introduced.

The experiments performed with various morphological elementary operators on synthetic and real images demonstrate the capabilities and restrictions of the novel approach.

Keywords: matrix field, tensor field, symmetric matrix, colour images, dilation, erosion, colour space, Einstein addition.

1 Introduction

Beginning with the path-breaking work of Matheron and Serra [15,16] in the late sixties mathematical morphology has provided us with an abundance of tools and techniques to process real valued-images for applications ranging from medical imaging to geological sciences [14,17,18,20]. Erosion and dilation are the fundamental operations of grey scale morphology relying on the notion of a minimum and a maximum of real numbers. Since minimum and maximum in

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 243–254, 2013.

turn depend on the presence of an order, it is no surprise that morphology for vector valued i.e. colour images does not always provide satisfactory results.

There have been numerous approaches how to extend the mathematical morphology framework to colour or vector-valued images. The main ingredients for such a framework are ranking schemes and the proper notion of extremal operators such as maximum and minimum. Due to the lack of reasonable complete lattice for vectorial data numerous suggestions for ranking schemes (based on various notions of distances, projections, and real-valued transforms) have been made, for a well structured, comprehensive, in-depth, and still up-to-date survey the reader is referred to [2] and the extensive list of literature cited therein. In [10] and [13] a more historic account is presented, while for a study of the background in order theory see [3] and [11]. Depending on the choices made one obtains morphological transforms with specific properties. However, none of these attempts seems to have been accepted unanimously in the image processing community. Somewhat surprisingly, the situation for (symmetric) matrix valued images is not as hopeless as it might seem at first glance. Here we consider a (symmetric) *matrix field* F as a mapping

$$F : \Omega \subset \mathbb{R}^d \longrightarrow \mathrm{Sym}(n)$$

from domain $\Omega \subset \mathbb{R}^d$ into the space $\mathrm{Sym}(n)$ of real symmetric $n \times n$-matrices with inner product $\langle A, B\rangle = \mathrm{trace}(AB)$ and (Frobenius-)norm $\|A\| = \sqrt{\langle A, A\rangle}$.

There have been successful attempts to extend the operations of mathematical morphology to images with values in the set of positive definite real symmetric 2×2- or 3×3-matrices [5,6,8] since these types of data make a natural appearance in medical imaging as the output of diffusion tensor weighted magnetic resonance imaging (DT-MRI).

The goal of this article is to present an approach to morphological operators for colour images by embedding a colour image suitably into a matrix field. Hence the morphology already developed for matrix fields will give rise to morphology for colour images. For the coding of a colour image as a matrix field we will make use of a variant of the HSL-colour space and the Loewner order cone for real symmetric 2×2-matrices. This novel concept can be applied to grey value images as well hence it includes scalar (flat) morphology.

However, we will not be able to decide the philosophical question what, for instance, the "right" maximum of the colors green and blue actually is. There is probably no unanimously acceptable answer, but an approach that recognizes that fact by producing the color white as that maximum for example is slightly more trustworthy than one producing different, so called "false" colors.

The structure of the article is as follows: In Section 2 we present the maximum and minimum operations for matrix-valued data and especially a three-dimensional representation of the Loewner order cone for 2×2- matrices. Section 3 deals with the aforementioned embedding and an operation for symmetric matrices gleaned from the relativistic addition of velocities. We report on results of our experiments with various morphological operators applied to synthetic and real colour images in Section 4. In Section 5 we offer concluding remarks and a glance at future research.

2 Loewner Ordering: Maximal and Minimal Matrices

The so-called *Loewner order* is a natural partial order on $\mathrm{Sym}(n)$, defined by means of the cone of positive semidefinite matrices $\mathrm{Sym}^+(n)$ by

$$A, B \in \mathrm{Sym}(n): \quad A \geq B :\Leftrightarrow A - B \in \mathrm{Sym}^+(n),$$

i.e. if and only if $A - B$ is positive semidefinite.

This partial order is *not* a lattice order, that is, there is no notion of a unique maximum and minimum with respect to this order [4]. Nevertheless, given any finite set of symmetric matrices $\mathcal{A} = \{A_1, \ldots, A_n\}$, we will be able to identify suitable maximal, resp., minimal matrices $\overline{A} := \max \mathcal{A}$ resp., $\underline{A} := \min \mathcal{A}$.

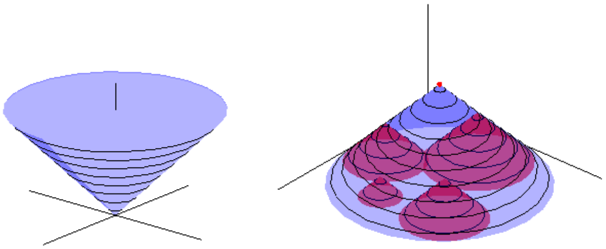
Since we will consider images with three colour components we may restrict ourselves from now on to the case of 2×2-matrices in $\mathrm{Sym}(2)$ which offer already three degrees of freedom. The procedure to find these extremal matrices for a set $\mathcal{A}$ is as follows: The cone $\mathrm{Sym}^+(2)$ can be represented in 3D using the bijection

$$\begin{pmatrix} \alpha \; \beta \\ \beta \; \gamma \end{pmatrix} \longleftrightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} 2\beta \\ \gamma - \alpha \\ \gamma + \alpha \end{pmatrix}, \text{ resp.}, \frac{1}{\sqrt{2}} \begin{pmatrix} z - y & x \\ x & z + y \end{pmatrix} \longleftrightarrow \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{1}$$

This linear mapping creates an isometrically isomorphic image of the cone $\mathrm{Sym}^+(2)$ in the Euclidean space $\mathbb{R}^3$ given by $\{(x, y, z)^\top \in \mathbb{R}^3 | \sqrt{x^2 + y^2} \leq z\}$ and is depicted in Fig. 1(a). For $A \in \mathrm{Sym}(2)$ the set $P(A) = \{Z \in \mathrm{Sym}(2) | A \geq Z\}$ denotes the penumbral cone or penumbra for short of the matrix A. It corresponds to a cone with vertex in A and a circular base in the $x - y$-plane: $P(A) \cap \{z = 0\} =$ circle with center $(\sqrt{2}\beta, \frac{\gamma - \alpha}{\sqrt{2}})$ and radius $\frac{\mathrm{trace}(A)}{\sqrt{2}}$.

Considering the associated penumbras of the matrices in $\mathcal{A}$ the search for the *maximal matrix* $\overline{A}$ amounts to determine the smallest penumbral cone covering all the penumbras of $\mathcal{A}$ tightly, see Fig. 1(b). One realises that the height of a penumbra measured from the $x - y$-plane is equal to the radius of its base, namely $\mathrm{trace}(A)/\sqrt{2}$. Hence a penumbra is already uniquely determined by the circle constituting its base. This implies that the search for a maximal matrix comes down to find the smallest circle enclosing the base-circles of the matrices in $\mathcal{A}$. This is a non-trivial problem in computer graphics. An numerical solution for finding the smallest circle enclosing the sampled basis circles has been implemented in C++ by Gärtner [12] and was used in [6] and [8]. However, in our case we employ the implementation of an efficient subgradient method detailed in [22] for the calculation of the smallest circle enclosing them. This gives us the smallest covering cone and hence the maximal matrix $\overline{A}$. To ensure that the base of the penumbra is located in the $x - y$-plane, we will not apply the above reasoning directly to the matrices $A_1, \ldots, A_n$, but to their shifted versions $\mathcal{A} + \kappa I := A_1 + \kappa I, \ldots, A_n + \kappa I$ with the unit matrix I and the parameter $\kappa := 1/\sqrt{2}$ taking advantage of the relation

$$\overline{\mathcal{A}} = \max(A_1, \ldots, A_n) = \max(A_1 + \kappa I, \ldots, A_n + \kappa I) - \kappa I = \overline{\mathcal{A} + \kappa I} - \kappa I\,.$$



(a) Loewner order cone (b) Covering cone

Fig. 1. Left: Image of the Loewner cone $\mathrm{Sym}^+(2)$. Right: Cone covering four penumbras of other matrices. The tip of each cone represents a symmetric 2×2 - matrix in $\mathrm{I\!R}^3$. For each cone the radius and the height are equal.

A suitable *minimal matrix* $\underline{A}$ is obtained by means of the formula

$$\underline{A} = \kappa I - \max(\kappa I - A_1, \ldots, \kappa I - A_n)$$

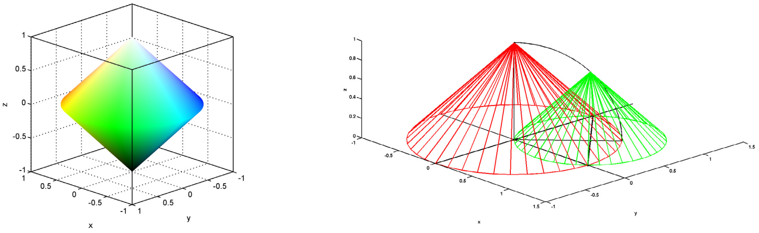
with unit matrix I and the parameter $\kappa = 1/\sqrt{2}$, inspired by the well-known scalar counterpart. For $i = 1, \ldots, n$ we have $\underline{A} \leq A_i \leq \overline{A}$ with respect to the Loewner order. We emphasise that $\overline{A}$ and $\underline{A}$ depend continuously on $A_1, \ldots, A_n$ by their construction. Also the rotational invariance is preserved, since the Loewner order is already rotational invariant: $A \geq B \iff UAU^\top \geq UBU^\top$ holds for any orthogonal matrix U. Nevertheless, the definitions of the matrices $\overline{A}$ and $\underline{A}$ are still meaningful for matrices that are not positive definite as long as they have a non-negative trace (since it corresponds to a radius in the construction above). It also becomes evident from their construction that in general neither $\overline{A}$ nor $\underline{A}$ coincide with any of the A_i: $\overline{A}, \underline{A} \notin \mathcal{A}$.

With these essential notions of suitable maximal and minimal matrices $\overline{A}$ and $\underline{A}$ at our disposal the definitions of the higher morphological operators carry over essentially verbatim.

3 Color Images as Matrix Fields

Closer to the human perception process than the well-known RGB color space is the HSL (or HSI) colour model describing a colour object by its hue, saturation and brightness resp. luminance (see [1, Algorithm 8.6.3] for the conversion).

Replacing in this model the coordinate saturation by the so-called *chroma* leads to a modified version of the HSL-model which we call HC$\tilde{\mathrm{L}}$ colour model. Note that we use this colour model due to its symmetry and its close relation to Ostwald's colour solid (a bicone) from 1916. The use of other colour models is close at hand as well. Its representation is given by a bicone C_2, depicted in Fig. 2(a). To be more specific: $\tilde{L} = 2L - 1$, for the chroma we have $C = \max\{R, G, B\} - \min\{R, G, B\}$. Hence, any point (x, y, z) of the bicone is coded



(a) Color bicone (b) The set of maximal and minimal matrices

Fig. 2. Left: $\mathrm{HC\tilde{L}}$-bicone, bicone for the $\mathrm{HC\tilde{L}}$ colour model. Right: The unit ball as the set of extremal matrices stemming from two bicone-matrices.

by $x = C\cdot \cos(2\pi\cdot H)$, $y = C\cdot \sin(2\pi\cdot H)$, $z = \tilde{L}$ providing us with an one-to-one transformation from the $\mathrm{HC\tilde{L}}$ colour space to the RGB colour space. The $\mathrm{HC\tilde{L}}$-bicone corresponds via (1) directly to the order interval $\frac{1}{\sqrt{2}}[-I, I]_L := \{A \in \mathrm{Sym}(2) \mid -\frac{1}{\sqrt{2}}I \leq A \leq \frac{1}{\sqrt{2}}I\}$, where the L abbreviates Loewner. In total this establishes the desired continuous one-to-one correspondence of the matrices in $\frac{1}{\sqrt{2}}[-I, I]_L$ with the colours in the $\mathrm{HC\tilde{L}}$ (and from there to the standard RGB space, if so desired),

$$\Psi : \mathrm{HC\tilde{L}} \subset I\!R^3 \longrightarrow \frac{1}{\sqrt{2}}[-I, I]_L \subset \mathrm{Sym}(2)\,.$$

Exploiting this correspondence one obtains, for example, the maximum of two colours $c_1, c_2 \in \mathrm{HC\tilde{L}}$ by transforming them into the matrices $\Psi(c_1), \Psi(c_2) \in \frac{1}{\sqrt{2}}[-I, I]_L$, then taking the maximum $\max(\Psi(c_1), \Psi(c_2))$ of these two matrices which is then transformed back to the new "maximal colour"

$$\max(c_1, c_2) := \Psi^{-1}\left(\max(\Psi(c_1), \Psi(c_2))\right)\,.$$

Ψ^{-1} combines the mapping (1), with the transform into polar coordinates via $H = \frac{1}{2\pi}\mathrm{arg(y,x)}\,, C = \sqrt{x^2 + y^2}\,, \tilde{L} = z\,,$ with a principal value of an appropriate argument function. The luminance L is obtained via $L = (\tilde{L} + 1)/2$ while the saturation is given by $S = 0$ if $C = 0$, otherwise $S = C/(1 - |2L - 1|)$. Having obtained those HSL-values, we convert them to the normalized RGB-values (see [1, Algorithm 8.6.4] for the conversion). The minimum of two colours is treated analogously. Hence by applying this rationale basic morphological operations can be transferred from matrix fields to color images.

However, we are facing two rather severe problems:

1. We have $\{\max(A, B) \cup \min(A, B)\ :\ A, B \in \frac{1}{\sqrt{2}}[-I, I]_L\} = I\!B_1$, that is, the set of vertices of cones supported by points from the bicone with respect to the Loewner order form the unit ball in $\mathrm{Sym}(n)$ visualised as the unit ball in $I\!R^3$. Hence corresponding HCL- and RGB-values do not exist. This can

be deduced with the help of Fig. 2, where the depicted green cone is extremal since it is supported and determined by the extreme points (=colors) $(1,0,0)$ and $(0,1,0)$. The claim follows by elementary geometric reasoning from varying these points on the unit circle combined with arguments of symmetry and rotational invariance.

2. Neither $\frac{1}{\sqrt{2}}[-I,I]_L$ nor $\mathbb{B}_1$ are closed under matrix addition, as it is necessary for the design of morphological top-hats or gradients. Again this would entail non-existence of corresponding HCL- and RGB-values.

We overcome the first difficulty by establishing a one-to-one mapping from the bicone C_2 to the unit ball $\mathbb{B}_1$. We begin by defining for a given point $(x,y,z) \in C_2$, a warp factor w by

$$w(x,y,z) = \begin{cases} 1, & \text{if } x,y = 0, \\ \frac{1+\Gamma}{\sqrt{1+\Gamma^2}} \text{ with } \Gamma = \frac{|z|}{\sqrt{x^2+y^2}}, & \text{otherwise.} \end{cases} \tag{2}$$

Multiplication of point coordinates with its warp factor establishes a one-to-one mapping from C_2 onto $\mathbb{B}_1$ in principle usable for the pull-back of matrices to the color cone C_2. However, experiments reveal that the use of this simple way mapping might lead to a fading of colors and the appearance of a grey tinge as reported in [7]. Instead we define a deforming warping factor

$$w_{\text{deform}}(x,y,z) = 1 + \left(\sqrt{x^2+y^2} + |z|\right)^n (w(x,y,z) - 1),$$

where $n \geq 1$ is some parameter. Large values of n cause points in the double-cone to remain almost unchanged, while the outer layer is not stretched proportionally. We choose $n = 10$ as it seems to be a large enough value to fit our needs. The mapping w_{deform} has to be inverted numerically by applying a root finding algorithm to

$$t^n - t^{n-1} + 1 - w_{\text{deform}}(x,y,z) = 0$$

for a point $(x,y,z) \in \mathbb{B}_1$. We always apply this inverse mapping (pull-back from $\mathbb{B}_1$ to C_2) after our procedures to find the extremal matrices, and prior the decoding of the matrices into colors. In total our "dilation" adopts the standard definition of the dilation but with respect to maximal color/matrix in the Loewner order and a subsequent pull-back in the sense described above. The same is true for our erosion.

To resolve the second problem we extend Einstein‘s general rule for the addition of velocities (see [19,21]) in the theory of Special Relativity (with speed of light $c = 1$) to symmetric matrices. Einstein's addition in $\mathbb{B}_1$ ensures that we stay inside $\mathbb{B}_1$ (this is our universe). More precisely, let $\alpha_A = \sqrt{1 - \|A\|^2}$ then define the Einstein addition $+_e$ by

$$A +_e B := \frac{1}{1 + \text{trace}(A \cdot B)} \left(A + \alpha_A \cdot B + \frac{1}{1+\alpha_A} \text{trace}(A \cdot B) \cdot A \right).$$

A commutative version $+_{ec} : I\!B_1 \times I\!B_1 \longrightarrow I\!B_1$, the Einstein coaddition (see [21]), reads in this context as

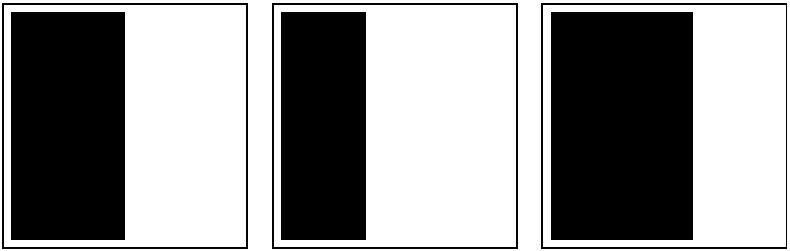
$$A +_{ec} B := \frac{\alpha_A B + \alpha_B A}{\alpha_A + \alpha_B} +_e \frac{\alpha_A B + \alpha_B A}{\alpha_A + \alpha_B} \,. \tag{3}$$

The corresponding subtraction is given by $A -_{ec} B := A +_{ec} (-B) = -B +_{ec} A$. Hence we can (Einstein-)add and subtract without restrictions in $I\!B_1$, as it is necessary for many morphological operators.

4 Experimental Results

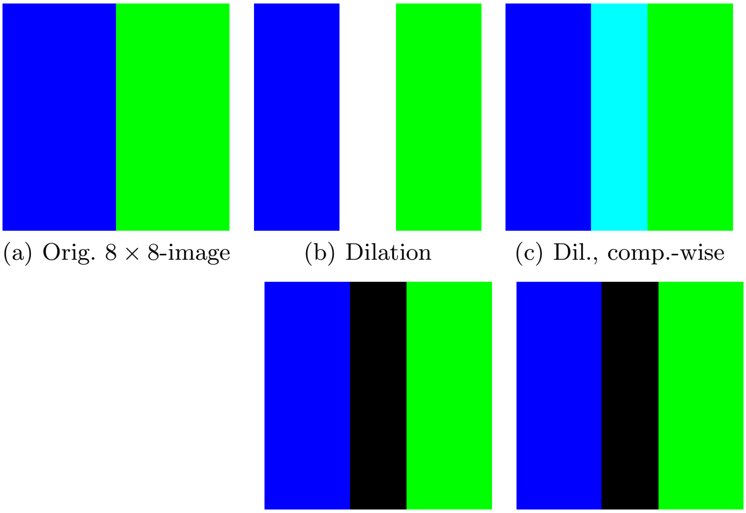
In this section some elementary morphological operators will be applied to both synthetic and natural colour images of various sizes. In the experiments we use different structuring elements (SE): a cross-shaped structuring element consisting of five pixels ($\mathrm{SE}_{\mathrm{cross}}$), a 3×3-square ($\mathrm{SE}_{3\times3}$), and a 11×11-square ($\mathrm{SE}_{11\times11}$), all centered at the middle pixel. Each of the images is extended by an appropriate layer of mirrored boundary values.

In the first experiments we confirm that our colour-morphological operators applied to image 3(a) in Fig. 3 in principle act as regular morphological operators on black-and-white images. As expected dilation and erosion result in an accurate shift of the inner object front, see Figs. 3(b), 3(c). Note that we surrounded the image by a black box to distinguish between the background of the image and the white background of the article. Again an image of size 8×8, but with a blue-coloured (RGB= $[0, 0, 255]$) left and a green-coloured (RGB= $[0, 255, 0]$) right is subjected to both a dilation and an erosion. Both colours are located in the $x - y$-plane relatively far apart on the boundary of the HC$\tilde{\mathrm{L}}$ bicone, hence the maximal and minimal matrices are representing white and black respectively. This accounts for the white resp. dark center section in the dilated resp. eroded images in Fig. 4(b) and Fig. 4(d). The component-wise approach produces “false” colors, as expected, at least for dilation. Now we consider an RGB-image of resolution 36×24 containing six cross-like structures with the RGB colours $[128, 255, 0]$, $[0, 128, 255]$, $[0, 0, 0]$, $[255, 255, 255]$, $[235, 249, 18]$ and



(a) Original 8×8-image (b) Dilation, $\mathrm{SE}_{\mathrm{cross}}$ (c) Erosion, $\mathrm{SE}_{\mathrm{cross}}$

Fig. 3. Dilation and erosion wih $\mathrm{SE}_{\mathrm{cross}}$ applied to bipartite black-and-white image


(a) Orig. 8 × 8-image (b) Dilation (c) Dil., comp.-wise

(d) Erosion (e) Ero., comp.-wise

Fig. 4. Dilation and erosion with $\mathrm{SE}_{\mathrm{cross}}$ applied to bipartite blue-and-green image

$[249, 155, 18]$ on a red background ($[255, 0, 0]$). Fig. 5(a) shows the constructed image on which a dilation with $\mathrm{SE}_{\mathrm{cross}}$ is applied. In Fig. 5(b) we see the result of the novel matrix-approach, whereas and Fig. 5(c) we see the results of a component-wise approach, namely the dilation of each of the RGB-channels separately. In both approaches the black and white crosses are processed equally well, however, for blue and green cross the channel-wise approach produces clearly false colors, while these crosses get a white halo in the matrix-approach. The

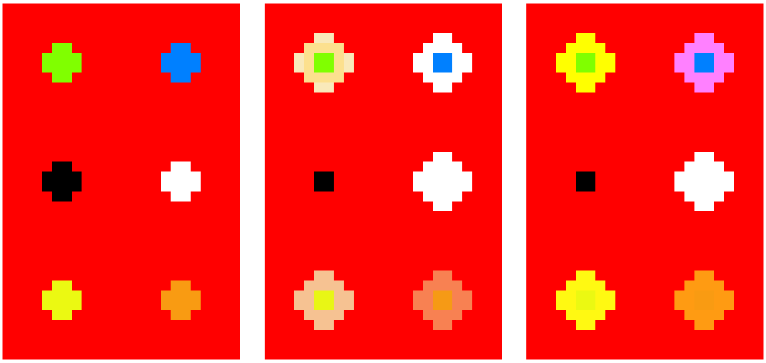
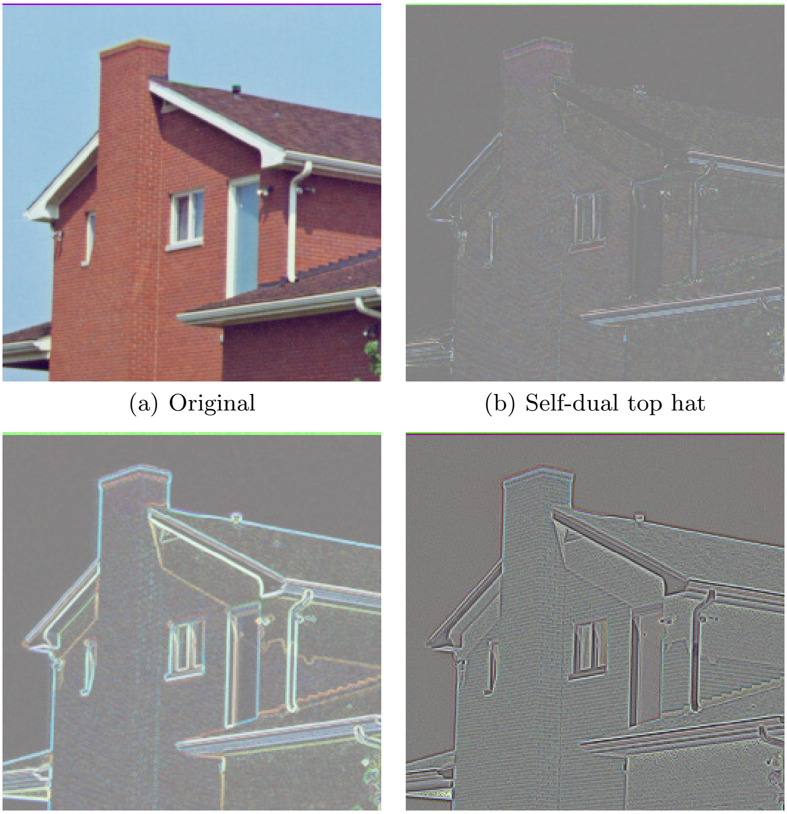

(a) Original (b) Erosion (c) Erosion, comp.-wise

Fig. 5. Erosion with $\mathrm{SE}_{\mathrm{cross}}$ applied to 36 × 24-synthetic image, in the matrix setting and component-wise

next set of experiments employs well-known morphological operators that involve taking the difference of two color images in the matrix setting. We restrict our attention to the self-dual top hat, the Beucher gradient, and the morphological Laplacian. Each operation uses the structuring element $\mathrm{SE}_{\mathrm{cross}}$. Taking the difference here means taking the difference in the sense of the aforementioned Einstein coaddition (3). As expected, in each of the cases the edges of the image objects become prominent, for the morphological derivatives more so than for the top hat. It is also no surprise that all the processed images have a strong grey tinge, the HSL-color close to the equatorial section of the bicone C_2.

It is important to remark that the artefacts in the processed house images are caused by an erroneous top pixel layer that are contained by default in the original image. In grey scale (flat) morphology there is the rule that iterated


(a) Original

(b) Self-dual top hat

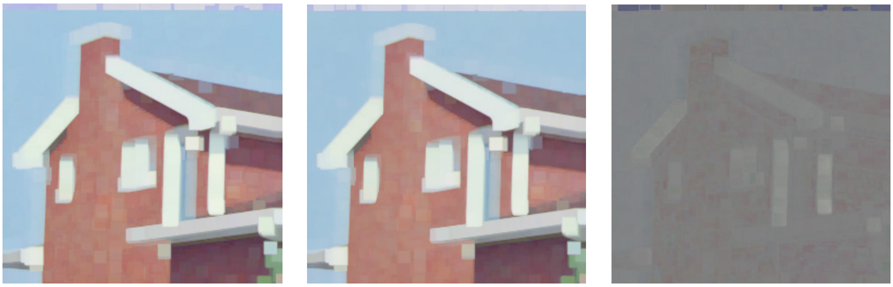
(c) Beucher grad.

(d) Morph. Laplacian

Fig. 6. Morphological operations with $\mathrm{SE}_{\mathrm{cross}}$ involving differences of images applied to the house-image (with a erroneous color at the top layer of pixels in the original). a) Original, b) self-dual top hat, c) Beucher gradient, d) morphological Laplacian.

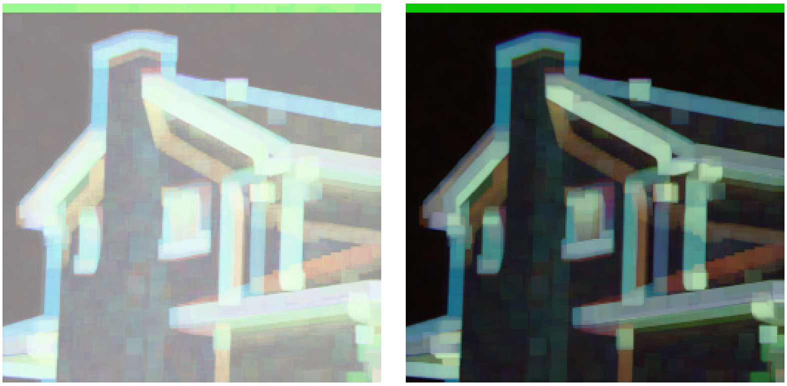
dilation with a structuring element amounts to the same as applying dilation once with the correspondingly dilated structuring element. For example, 5-fold dilation with a 3 × 3-square should produce the same result as a single dilation with a 11 × 11-square. In our matrix-approach this cannot be expected to be the case. We apply to the house image first a dilation with a 11 × 11-square, then a 5-fold dilation with a 3 × 3-square. The (Einstein-)difference image is displayed as well in Fig. 7. We again compare the matrix-approach to the RGB-channel-wise approach, this time for a natural image in the case of the Beucher gradient for a greater effect. In both cases the edge-detecting capabilities of the Beucher gradient are prominent. A difference between the results is noticeable as one can see in Figure 8, but it is now difficult to judge which is better than the other.

Finally note that we did all computation with Matlab on a regular PC. We need 1.1, 4.0, 15.7, 62.2, and 276.2 seconds for a dilation or erosion of an image of resolution 32 × 32, 64 × 64, 128 × 128, 256 × 256, and 512 × 512, respectively.



(a) Dilation, 11 × 11-SE (b) Dilation, 3×3-SE, 5-fold (c) Difference

Fig. 7. Dilation applied to house-image: a) with 11 × 11-SE, b) with a 3 × 3-SE performed 5 times, c) non-zero difference image (Einstein subtraction)



(a) Beucher, 11 × 11-SE (b) Beucher, 11 × 11-SE, comp.-wise

Fig. 8. Beucher Gradient applied to house-image with 11 × 11-SE, a) matrix setting, b) component-wise

Using the profile tool of Matlab reveals that the most time consuming part is needed by the subgradient method to find the smallest enclosing circle of a set of circles. A further speedup will only be possible using C++.

5 Conclusion

Real symmetric 2×2-matrices offer three degrees of freedom. Hence the three components of many popular colour models, such as the RGB- or the HSI-model can be coded as such a matrix. The matrix-setting has indeed several advantages over the vector-setting of colour images: First the algebraic structure is richer, second, there exists a canonical order, the Loewner order, for symmetric matrices. This coding was inspired by the close, almost obvious geometric relation between the $\widetilde{\text{HCL}}$ bicone, a variant of the HSI-bicone, and the order interval $\frac{1}{\sqrt{2}}[-I, I]_L$ induced by the Loewner order cone. As soon as the colour image has been rewritten in this way, the morphological techniques developed for matrix fields in [8] and [9] were in principle applicable. However the bicone is not ideal for computations. We used a one-to-one nonlinear mapping to go from the bicone C_2 to the unit ball $\mathbb{B}_1$ in the set of symmetric 2×2 matrices and back, and combined our matrix-valued maximum/minimum operation with it. Standard addition leading to values outside an order interval have been replaced by Einstein coaddition under which $\mathbb{B}_1$ is stable. With these novel techniques various morphological operations ranging from the elementary dilation and erosion to second order derivatives could be studied. The experimental results are promising and may serve as a proof-of-concept for this novel approach to colour image morphology. It is clear that the choice of the color cone has a great influence on the results, and in the future will explore other color models as the basis for this approach to morphology for color or even multispectral images. A more in-depth comparison with other existing color morphology models is a subject of future research.

References

1. Agoston, M.K.: Computer Graphics and Geometric Modeling: Implementation and Algorithms. Springer, London (2005)
2. Aptoula, E., Lefèvre, S.: A comparative study on multivariate mathematical morphology. Pattern Recognition 40(11), 2914–2929 (2007)
3. Barnett, V.: The ordering of multivariate data. Journal of the Statistical Society, A 139(3), 318–355 (1976)
4. Borwein, J.M., Lewis, A.S.: Convex Analysis and Nonlinear Optimization. Springer, New York (2000)
5. Brox, T., Weickert, J.: A TV flow based local scale measure for texture discrimination. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3022, pp. 578–590. Springer, Heidelberg (2004)
6. Burgeth, B., Bruhn, A., Papenberg, N., Welk, M., Weickert, J.: Mathematical morphology for tensor data induced by the Loewner ordering in higher dimensions. Signal Processing 87(2), 277–290 (2007)

7. Burgeth, B., Kleefeld, A.: Order based morphology for color images via matrix fields. In: Burgeth, B., Vilanova, A., Westin, C.-F. (eds.) Visualization and Processing of Tensor Fields and Higher Order Descriptors for Multi-Valued Data. Springer, Berlin (submitted)
8. Burgeth, B., Papenberg, N., Bruhn, A., Welk, M., Feddern, C., Weickert, J.: Mathematical morphology based on the loewner ordering for tensor data. In: Ronse, C., Najman, L., Decencière, E. (eds.) Mathematical Morphology: 40 Years On. Computational Imaging and Vision, vol. 30, pp. 407–418. Springer, Dordrecht (2005)
9. Burgeth, B., Welk, M., Feddern, C., Weickert, J.: Mathematical morphology on tensor data using the Loewner ordering. In: Weickert, J., Hagen, H. (eds.) Visualization and Processing of Tensor Fields, pp. 357–367. Springer, Berlin (2006)
10. Comer, M.L., Delp, E.J.: Morphological operations for color image processing. Journal of Electronic Imaging 8(3), 279–289 (1999)
11. Dougherty, E.R. (ed.): Anamorphoses and function lattices, Mathematical Morphology in Image Processing, pp. 483–523. Marcel Dekker, New York (1993)
12. Gaertner, B.: Smallest enclosing balls of points - fast and robust in C++, `http://www.inf.ethz.ch/personal/gaertner/miniball.html` (last visited July 03, 2012)
13. Goutsias, J., Heijmans, H.J.A.M., Sivakumar, K.: Morphological operators for image sequences. Computer Vision and Image Understanding 62, 326–346 (1995)
14. Heijmans, H.J.A.M.: Morphological Image Operators. Academic Press, Boston (1994)
15. Matheron, G.: Eléments pour une théorie des milieux poreux. Masson, Paris (1967)
16. Serra, J.: Echantillonnage et estimation des phénomènes de transition minier. PhD thesis, University of Nancy, France (1967)
17. Serra, J.: Image Analysis and Mathematical Morphology, vol. 1. Academic Press, London (1982)
18. Serra, J.: Image Analysis and Mathematical Morphology, vol. 2. Academic Press, London (1988)
19. Sexl, R.U., Urbantke, H.K.: Relativity, Groups, Particles: Special Relativity and Relativistic Symmetry in Field and Particle Physics. Springer, Wien (2001)
20. Soille, P.: Morphological Image Analysis, 2nd edn. Springer, Berlin (2003)
21. Ungar, A.A.: Einstein's special relativity: The hyperbolic geometric viewpoint. In: Conference on Mathematics, Physics and Philosophy on the Interpretations of Relativity, II, Budapest (2009)
22. Xu, S., Freund, R.M., Sun, J.: Solution Methodologies for the Smallest Enclosing Circle Problem. In: High Performance Computation for Engineered Systems (HPCES) (2003), http://hdl.handle.net/1721.1/4015

A Multivariate Mathematical Morphology Based on Orthogonal Transformation, Probabilistic Extrema Estimation and Distance Optimization

Alexandru Căliman[1], Mihai Ivanovici[1], Nöel Richard[2], and Gheorghe Toacşe[1]

[1] MIV^2 Laboratory
Department of Electronics and Computers, Transilvania University
Braşov, România
[2] XLIM-SIC UMR CNRS 6172
Signals, Images and Communications, University of Poitiers
Poitiers, France

Abstract. Mathematical morphology (MM) is a very popular image processing framework, which offers widely-used non-linear tools. It was introduced for binary and greylevel images, but recently, numerous approaches have been proposed for color or multivariate images. Many of these approaches are based on the lexicographical ordering, which respects the total ordering properties, thus making this approach a very robust solution. However, it also has disadvantages like the subjective prioritization of the components and the perceptual nonlinearities introduced due to color component prioritization. Within this paper, we introduce a new multivariate MM approach, derived from a probabilistic approach, through the optimization of the distance between the estimated pseudo-extrema and vectors within the initial data set. We compare the results generated using the two approaches and a generic lexicographic approach based on Principal Component Analysis as the axis prioritization criteria.

Keywords: Multivariate Mathematical Morphology, Principal Component Analysis, Color Image Segmentation.

1 Introduction

The mathematical morphology (MM) was introduced by Matheron in the seventh decade of the last century, as a non-linear processing technique for binary images [1] and it subsequently became more popular in the following decade, through Serra's reference writings [2] [3]. The extension to greylevel images, through the *umbra* or *cross-sections* concepts [4] [5] allowed a generalization which led to the appearance of many image processing and analysis algorithms based on this framework e.g. filters [6], image segmentation [7], texture description [8], etc. The greylevel MM (GLMM) is based on the lattice theory, which implies a partial ordering of the image data. Therefore, the existence of an *infimum* and a *supremum* for any subset of the pixel data set is sufficient in order to define a MM upon the given set [9]. For the extension to multivariate images, there have been proposed several approaches. In [10], four types of vectorial orderings have been proposed (marginal, reduced, conditional and partial), but only

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 255–266, 2013.

the conditional and the marginal orderings generate unique extrema, accomplishing the anti-symmetry property[1]. However, within the marginal approach there exist vectors which are not comparable and thus, when the marginal ordering is used for computing the two extrema of a given set, its main disadvantage arise: it introduces *false colors* i.e. data which does not belong to the initial image distribution [11]. In addition, the conditional ordering generates a non-linear ordering from the human visual system point of view e.g. in the RGB colorspace, for a (red, green, blue) fixed priority, with 8 bits per channel, $(0, 0, 255)$ 🟦 $<$ $(0, 255, 0)$ 🟩 $<$ $(1, 0, 255)$ 🟦 although, from a human perception point of view, the two blue shades are practically identical. Therefore, depending on application, the questions of non-ambiguity and pertinence of the ordering arise. There have been proposed a plethora of methods for color and multivariate MM, especially based on the marginal, reduced and conditional orderings, but none of them has been imposed as a generally-valid solution [12]. Recently, an approach based on the partial ordering has also been introduced, although, in order to accomplish the anti-symmetry property, the conditional ordering is also implied [13] [14]. There are also approaches which avoid defining an underlying ordering relation among the image data and only find *infimum* and *supremum* for the given vectorial data [15] [16]. An example of such an approach uses the distribution of the data, combined with the Chebychev's inequality in order to estimate the two extrema for the multivariate data [17]. Even if it is not a marginal approach, this method generates false colors due to the manner in which these extrema are chosen i.e. as vectorial data located on the first principal component, generated through principal component analysis (PCA) technique. The extrema are chosen according to the Tchebychev's inequality which generates two bounds, on one side and on the other of the multivariate data mean, along the first principal component. In this paper, we introduce a new multivariate MM approach derived from this method, in order to generate data which belongs to the initial image distribution. First, we describe a simple lexicographic ordering defined within the Principal Component Analysis (PCA) space. Next, we briefly describe the probabilistic mathematical morphology introduced in [17], and finally we derive the proposed approach through the optimization of the distance between the estimated extrema and the vectors within the initial data set. Our first aim is to correct the false color issue by choosing the image data from the initial set, which are the closest from the estimated extrema in terms of perceptual distance. Secondly, we compare the presented approaches from a qualitative point of view, emphasizing the advantages of using the local distributions in front of the global lexicographic approach.

2 Finding the Optimum Space Representation for the Image Data

In [18] a maximum noise fraction (MNF) decorrelating transform is used in order to find an optimum space representation before marginal ordering. Also, in [17], a PCA transform is applied on the entire image data, in order to define the global references which are subsequently used for the local extrema ordering. However, in that case, the local vectors are not ordered according to the global PCA axis, but two local extrema are generated from each local distribution. These local extrema may not belong to the initial

[1] The conditional ordering generates unique extrema only when all the data components are used.

set, thus generating false colors. PCA finds the orthogonal directions on which the variance of the data is maximized, so it can be used as an optimum space for representing the vectorial image data. Our first objective is to apply a lexicographical ordering for every local data set [19], based on the global axis generated through a PCA for the entire image. Consequently, we propose a way in which the priorities may be chosen i.e. the first principal component, which has the largest variance and thus embeds the most information, is chosen to be the most important while the last principal component receives the last priority. Therefore, given a set $\mathcal{S}$ of local image vectors $\mathbf{v_k} = \big(v_{k_1}, v_{k_2}, \ldots, v_{k_n}\big)$ represented in global PCA coordinates as $\widetilde{\mathbf{v_k}} = \big(\widetilde{v_{k_1}}, \widetilde{v_{k_2}}, \ldots, \widetilde{v_{k_n}}\big)$, then the strict lexicographic order within the PCA space is defined as:

$$\mathbf{v_i} < \mathbf{v_j} \Leftrightarrow \begin{cases} \widetilde{v_{i_1}} < \widetilde{v_{j_1}} \quad \text{or} \\ \widetilde{v_{i_1}} = \widetilde{v_{j_1}} \quad \text{and} \ \widetilde{v_{i_2}} < \widetilde{v_{j_2}} \quad \text{or} \\ \qquad \cdots \\ \widetilde{v_{i_1}} = \widetilde{v_{j_1}} \quad \text{and} \ \widetilde{v_{i_2}} = \widetilde{v_{j_2}} \quad \text{and} \ \cdots \ \text{and} \ \widetilde{v_{i_n}} < \widetilde{v_{j_n}} \end{cases} \tag{1}$$

In addition, if an ordering relation is defined as $(\mathbf{v_i} \leq \mathbf{v_j}) \Leftrightarrow ((\mathbf{v_i} < \mathbf{v_j}) \ \text{or} \ (\mathbf{v_i} = \mathbf{v_j}))$ then, within a finite multivariate data set $\mathcal{S}$, we may define two extrema, the minimum (*min*) and maximum (*max*), of the given set as:

$$\begin{cases} min\big(\mathcal{S}\big) = \{\mathbf{v_i} \in \mathcal{S} \mid \mathbf{v_i} \leq \mathbf{v_k}, \forall \mathbf{v_k} \in \mathcal{S}\} \\ max\big(\mathcal{S}\big) = \{\mathbf{v_i} \in \mathcal{S} \mid \mathbf{v_k} \leq \mathbf{v_i}, \forall \mathbf{v_k} \in \mathcal{S}\} \end{cases} \tag{2}$$

Using the defined ordering, then for every nonempty vectorial set $\mathcal{S}$ there exist unique minimum and maximum values. Therefore, a MM framework based on flat structuring elements (SE) may be introduced for multivariate images. Thus, given an image $f : \mathcal{D}_f \rightarrow \mathcal{S}_{\mathcal{D}f}$, with the support $\mathcal{D}_f \subset \mathbb{Z}^2$ and codomain $\mathcal{S}_{\mathcal{D}f} \subset \mathbb{R}^n$ and a flat SE g with the support $\mathcal{D}_g \subset \mathbb{Z}^2$, the erosion and the dilation, the two fundamental morphological operations are defined as [20]:

$$[\varepsilon_g(f)](x) = \min_{z \in \mathcal{D}_g} f(x+z), \ \ \forall x \in \mathcal{D}_f \tag{3}$$

$$[\delta_g(f)](x) = \max_{z \in \mathcal{D}_g} f(x+z), \ \ \forall x \in \mathcal{D}_f \tag{4}$$

The results generated using this approach are presented in section 5. Given the lexicographical approaches disadvantages, e.g. generating perceptual nonlinearities and the subjective choice of the components priorities, we proposed a probabilistic multivariate MM, in which the local data are not ordered, but two *pseudo*-extrema are generated in a probabilistic way for each local color set within a multivariate image [17]. Next, we briefly describe this approach.

3 Probabilistic Mathematical Morphology

In this section, we briefly describe the probabilistic mathematical morphology (PMM), which was introduced in [17]. This approach is based on the Tchebychev inequality [21] which allows us to predict, for any distribution, the upper limit of the probability to have data in a given symmetric interval around the mean:

$$P\{|\xi - \mu_\xi| \geq k\sigma_\xi\} \leq \frac{1}{k^2}. \tag{5}$$

where ξ is a random variable with finite mean μ_ξ and standard deviation σ_ξ and k is a constant value. With the help of the k parameter, symmetrical intervals around the mean value can be generated. The bounds of this interval are considered to be the probabilistic extrema $\mathcal{E}_\alpha$ and $\mathcal{E}_\beta$, in the sense of the Tchebychev inequality and are defined as:

$$\begin{cases} \mathcal{E}_\alpha = \mu_\xi + k\sigma_\xi \\ \mathcal{E}_\beta = \mu_\xi - k\sigma_\xi \end{cases} \tag{6}$$

with $\mathcal{E}_\alpha$ as the *pseudo-supremum* and $\mathcal{E}_\beta$ as the *pseudo-infimum* of the given distribution. According to the k parameter these pseudo-extrema are closer or farther from the real maximum or minimum. For symmetrical distributions there exist a unique k value for which the pseudo-extrema coincide with real extrema. Otherwise, for a given k, one pseudo-extremum will be identical to the corresponding real extremum, while for the other there will be an estimation error.

Using these notions, a MM framework based on flat SEs may be introduced for greyscale images. Thus, for an image $f : \mathcal{D}_f \to \mathcal{S}_{\mathcal{D}f}$, with the codomain $\mathcal{S}_{\mathcal{D}f} \subset \mathbb{R}$ and a flat SE g with the support $\mathcal{D}_g$, the erosion and the dilation, are defined as:

$$[\varepsilon_g(f)](x) \overset{\text{Def}}{=} \bigwedge_{z \in \mathcal{D}_g} f(x+z) = \mu_\xi - k\sigma_\xi, \ \ \forall x \in \mathcal{D}_f \tag{7}$$

$$[\delta_g(f)](x) \overset{\text{Def}}{=} \bigvee_{z \in \mathcal{D}_g} f(x+z) = \mu_\xi + k\sigma_\xi, \ \ \forall x \in \mathcal{D}_f \tag{8}$$

where $\wedge$ and $\vee$ are the *pseudo-infimum* and *pseudo-supremum* operators and ξ represents the particular outcome of a random variable, modeling the intensity of the greyscale image data given by $\mathcal{D}_f \cap \mathcal{D}_g$. Within this construction, the k parameter may be used to adjust the estimation error: a small k value would generate pseudo-extrema close to the mean value of the given set, while the larger the k, the farther the pseudo-extrema would be from the mean. In other words, a small value of k leads to operations with an almost linear behavior, while as the k is increased, the operations behavior becomes more non-linear, specific to the morphological operations. In Fig. 1 the classical GLMM using flat SE is compared with the greyscale PMM, for various k values.

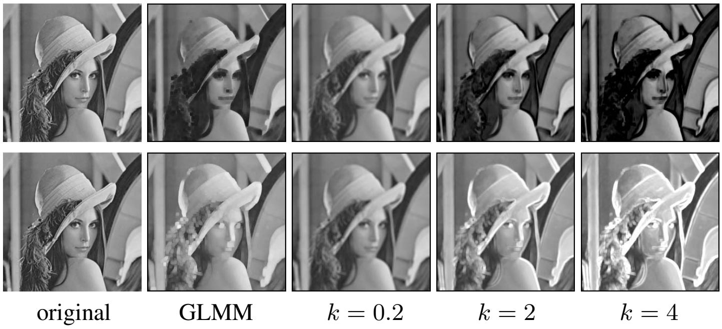


Fig. 1. Erosions (top row) and dilations (bottom row), 5×5 SE size, GLMM and PMM

A quantification of the total error in extrema estimation, with respect to the real extrema as a function of k and a more detailed description and validation of the PMM approach is presented in [22].

The approach was extended to multivariate images by applying the Tchebychev inequality on the first principal component of the local data, generated by PCA. Thus, given a set $\mathcal{S} \subset \mathbb{R}^n$ of multivariate data $\mathbf{v_x} = \left(v_{x_1}, v_{x_2}, \ldots, v_{x_n}\right)$, which after the PCA transform has the coordinate $\widetilde{v_{x_i}}$ on the ith principal component, the two probabilistic pseudo-extrema of the set are defined within the PCA axis system as:

$$\begin{cases} \widetilde{\mathcal{E}_\alpha} = \left(\mu_{\widetilde{v_{x_1}}} + k\sigma_{\widetilde{v_{x_1}}}, 0, 0, \cdots\right) \\ \widetilde{\mathcal{E}_\beta} = \left(\mu_{\widetilde{v_{x_1}}} - k\sigma_{\widetilde{v_{x_1}}}, 0, 0, \cdots\right) \end{cases} \tag{9}$$

in which $\mu_{\widetilde{v_{x_1}}}$ and $\sigma_{\widetilde{v_{x_1}}}$ represents the mean and the standard deviation of the first principal component of the multivariate data $\{\mathbf{v_x}\} \subset \mathcal{S}$. The vectorial probabilistic extrema are obtained by representing $\widetilde{\mathcal{E}_\alpha}$ and $\widetilde{\mathcal{E}_\beta}$ in the initial coordinates system, thus resulting $\mathcal{E}_\alpha$ and $\mathcal{E}_\beta$. However, the sign associated with the extrema in the PCA basis cannot reflect their order, because the PCA is performed through data rotations, which can be performed either clockwise or counter-clockwise. In order to establish which of the two extrema is associated with the morphological erosion or with the dilation, they are ordered lexicographically, according to n pairs of reference points. The references are positioned on perpendicular lines, thus forming an orthogonal axis system and they could be either manually or automatically chosen. In this paper, in the particular case of color images, we chose to compute the three pairs of references as the pseudo-extrema of the entire image data distribution. Furthermore, they are ordered, pair by pair, by projection on the black-white axis, thus generating an orthogonal axis system. Therefore, generalizing to the n-dimensional case, with $\left(\mathcal{R}_0^-, \mathcal{R}_0^+\right)$, $\left(\mathcal{R}_1^-, \mathcal{R}_1^+\right)$, $\left(\mathcal{R}_2^-, \mathcal{R}_2^+\right), \cdots$ as ordered reference points, the two morphological operations are defined as:

$$[\varepsilon_g(f)](x) = \bigwedge_{\substack{z \in \mathcal{D}_g \\ x \in \mathcal{D}_f}} f(x+z) \overset{\text{Def}}{=} \begin{cases} \arg\min\limits_i [\overrightarrow{\mathcal{R}_0^- \mathcal{R}_0^+} \cdot \overrightarrow{\mathcal{R}_0^- i}] \text{ , with } i \in \{\mathcal{E}_\alpha, \mathcal{E}_\beta\} \\ \arg\min\limits_i [\overrightarrow{\mathcal{R}_1^- \mathcal{R}_1^+} \cdot \overrightarrow{\mathcal{R}_1^- i}] \text{ , with } i \in \{\mathcal{E}_\alpha, \mathcal{E}_\beta\} \\ \qquad\qquad\qquad\qquad \text{if } \overrightarrow{\mathcal{R}_0^- \mathcal{R}_0^+} \cdot \overrightarrow{\mathcal{E}_\alpha \mathcal{E}_\beta} = 0 \\ \arg\min\limits_i [\overrightarrow{\mathcal{R}_2^- \mathcal{R}_2^+} \cdot \overrightarrow{\mathcal{R}_2^- i}] \text{ , with } i \in \{\mathcal{E}_\alpha, \mathcal{E}_\beta\} \\ \quad \text{if } \overrightarrow{\mathcal{R}_0^- \mathcal{R}_0^+} \cdot \overrightarrow{\mathcal{E}_\alpha \mathcal{E}_\beta} = 0 \text{ and } \overrightarrow{\mathcal{R}_1^- \mathcal{R}_1^+} \cdot \overrightarrow{\mathcal{E}_\alpha \mathcal{E}_\beta} = 0 \\ \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \end{cases} \tag{10}$$

$$[\delta_g(f)](x) = \bigvee_{\substack{z \in \mathcal{D}_g \\ x \in \mathcal{D}_f}} f(x+z) \overset{\text{Def}}{=} \begin{cases} \arg\max\limits_i [\overrightarrow{\mathcal{R}_0^- \mathcal{R}_0^+} \cdot \overrightarrow{\mathcal{R}_0^- i}] \text{ , with } i \in \{\mathcal{E}_\alpha, \mathcal{E}_\beta\} \\ \arg\max\limits_i [\overrightarrow{\mathcal{R}_1^- \mathcal{R}_1^+} \cdot \overrightarrow{\mathcal{R}_1^- i}] \text{ , with } i \in \{\mathcal{E}_\alpha, \mathcal{E}_\beta\} \\ \qquad\qquad\qquad\qquad \text{if } \overrightarrow{\mathcal{R}_0^- \mathcal{R}_0^+} \cdot \overrightarrow{\mathcal{E}_\alpha \mathcal{E}_\beta} = 0 \\ \arg\max\limits_i [\overrightarrow{\mathcal{R}_2^- \mathcal{R}_2^+} \cdot \overrightarrow{\mathcal{R}_2^- i}] \text{ , with } i \in \{\mathcal{E}_\alpha, \mathcal{E}_\beta\} \\ \quad \text{if } \overrightarrow{\mathcal{R}_0^- \mathcal{R}_0^+} \cdot \overrightarrow{\mathcal{E}_\alpha \mathcal{E}_\beta} = 0 \text{ and } \overrightarrow{\mathcal{R}_1^- \mathcal{R}_1^+} \cdot \overrightarrow{\mathcal{E}_\alpha \mathcal{E}_\beta} = 0 \\ \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \end{cases} \tag{11}$$

in which $\mathcal{E}_\alpha$ and $\mathcal{E}_\beta$ are the extrema of the local data given by $\mathcal{D}_f \cap \mathcal{D}_g$. Eq (10) and (11) represent the lexicographical ordering of the local extrema, using the PCA space computed for the entire image data. However, these extrema do not belong to the initial set, thus generating false colors (see the results in section 5). In order to solve this problem, we propose an adaptation of this method, through the optimization of the distance between the estimated extrema and the vectorial data within the initial set.

4 False Color Problem Solving

Our aim is to find the vectors within the initial data set, for which the distance between the respective vectors and the estimated extrema is minimum, while the k parameter, which reflects the distance between the two local extrema, is maximum. Therefore, we consider a set $\mathcal{S} \subset \mathbb{R}^n$ of multivariate data $\mathbf{v_x}$, which after PCA has the coordinates $\widetilde{\mathbf{v_x}} = \big(\widetilde{v_{x_1}}, \widetilde{v_{x2}}, \ldots, \widetilde{v_{x_n}}\big)$, with $\widetilde{v_{x_i}}$ as the $\mathbf{v_x}$ coordinate on the ith principal component. The quantities which have to be minimized are $dist\big(\mathbf{v_i}, \mathcal{E}_\alpha\big)$ and $dist\big(\mathbf{v_j}, \mathcal{E}_\beta\big)$ for $\mathbf{v_i}, \mathbf{v_j} \in \mathcal{S}$, while $dist\big(\mathcal{E}_\alpha, \mathcal{E}_\beta\big)$ has to be maximum (see Fig. 2). As the $dist(\cdot,\cdot)$ distance, we used the Euclidean (L2) norm. Therefore, considering that $\mu_{\widetilde{v_{x_i}}} = 0$ for all the PCA generated components,

$$\begin{cases} dist\big(\mathbf{v_i}, \mathcal{E}_\alpha\big) &= \sqrt{(k\sigma_{\widetilde{v_{x_1}}} - \widetilde{v_{i_1}})^2 + \widetilde{v_{i_2}}^2 + \cdots + \widetilde{v_{i_n}}^2} \\ dist\big(\mathbf{v_j}, \mathcal{E}_\beta\big) &= \sqrt{(-k\sigma_{\widetilde{v_{x_1}}} - \widetilde{v_{j_1}})^2 + \widetilde{v_{j_2}}^2 + \cdots + \widetilde{v_{j_n}}^2} \\ dist\big(\mathcal{E}_\alpha, \mathcal{E}_\beta\big) &= k\sigma_{\widetilde{v_{x_1}}} - (-k\sigma_{\widetilde{v_{x_1}}}) = 2k\sigma_{\widetilde{v_{x_1}}} \end{cases} \tag{12}$$

We face the delicate issue of maximizing a linear function, with two given constraints, which are not well-defined. Consequently, the feasibility of this optimization problem will be the subject of a future paper. However, here we propose an algorithmic solution to this problem. Hence, we aim in finding the k values for which the two distances are minimum, so we impose $\frac{d}{dk}\big(dist\big(\mathbf{v_i}, \mathcal{E}_\alpha\big)\big) = 0$ and $\frac{d}{dk}\big(dist\big(\mathbf{v_j}, \mathcal{E}_\beta\big)\big) = 0$ which means solving the following system:

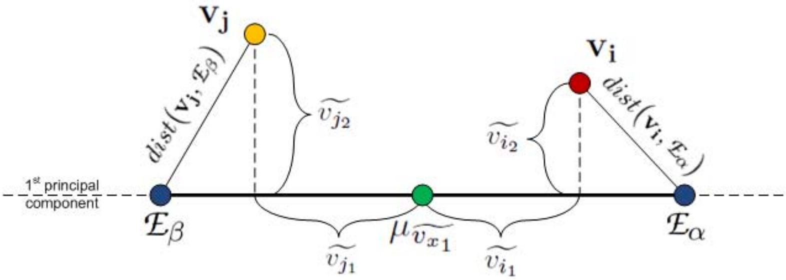


Fig. 2. PCA coordinates for a 2-dimensional case

$$\begin{cases} \dfrac{\sigma_{\widetilde{v_{x_1}}}\left(k\sigma_{\widetilde{v_{x_1}}}-\widetilde{v_{i_1}}\right)}{\sqrt{\left(k\sigma_{\widetilde{v_{x_1}}}-\widetilde{v_{i_1}}\right)^2+\widetilde{v_{i_2}}^2+\dots+\widetilde{v_{i_n}}^2}} = 0 \\ \dfrac{-\sigma_{\widetilde{v_{x_1}}}\left(-k\sigma_{\widetilde{v_{x_1}}}-\widetilde{v_{j_1}}\right)}{\sqrt{\left(-k\sigma_{\widetilde{v_{x_1}}}-\widetilde{v_{j_1}}\right)^2+\widetilde{v_{j_2}}^2+\dots+\widetilde{v_{j_n}}^2}} = 0 \end{cases} \tag{13}$$

There is a unique k value that satisfies the both equations if and only if $\widetilde{v_{i_1}} = -\widetilde{v_{j_1}}$. Otherwise, a unique solution k will definitely lead to the appearance of at least one false color within the morphological process. In our particular case, there are two values: $k = k_1$ which satisfies the first equation and $k = k_2$ for the second one, where

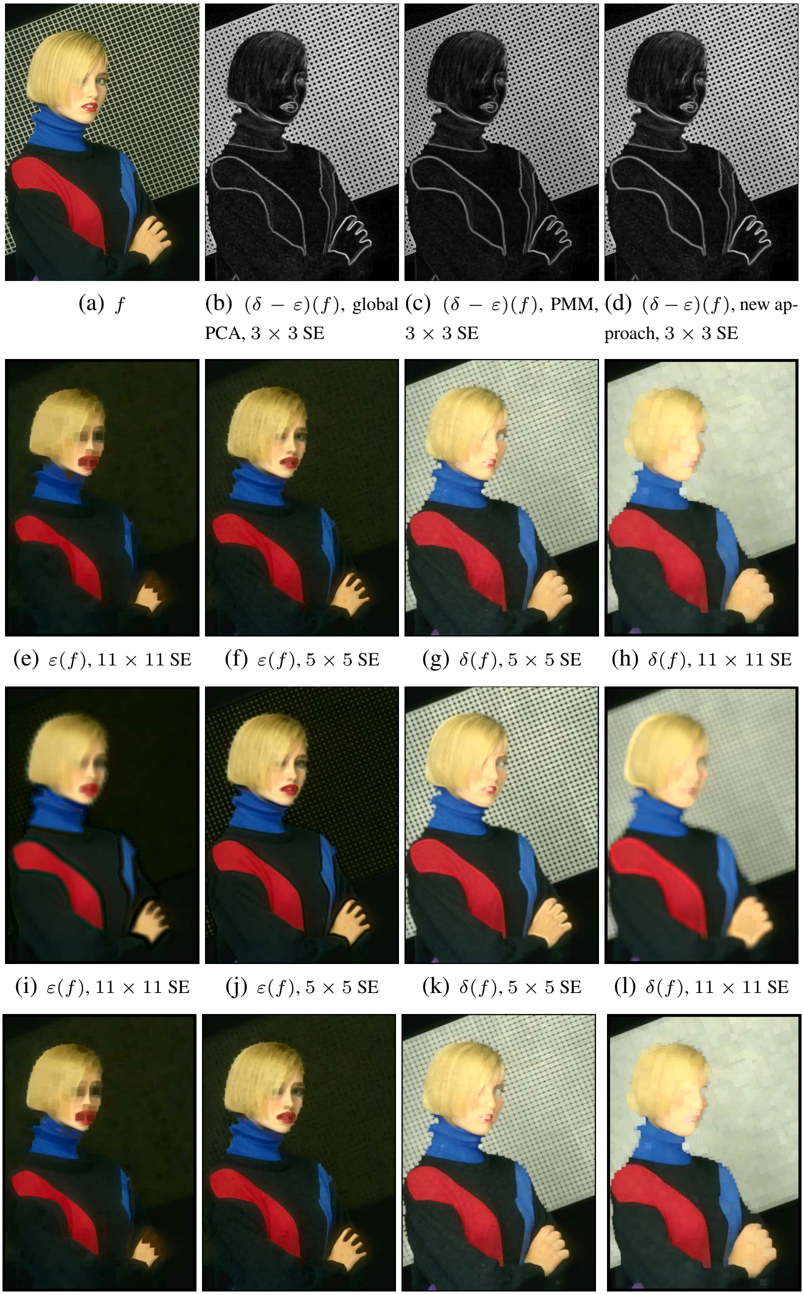
$$k_1 = \frac{\widetilde{v_{i_1}}}{\sigma_{\widetilde{v_{x_1}}}}, \quad k_2 = \frac{-\widetilde{v_{j_1}}}{\sigma_{\widetilde{v_{x_1}}}} \tag{14}$$

The interpretation of these solutions is that k_1 and k_2 are the corresponding k values for the first principal components of the $\mathbf{v_i}$ and $\mathbf{v_j}$ vectors. However, we also aim in maximizing $dist\left(\mathcal{E}_\alpha, \mathcal{E}_\beta\right)$. Therefore, the resulting extrema vectors are $\mathbf{v_i}$, $\mathbf{v_j} \in \mathcal{S}$ with the property that $\widetilde{v_{i_1}} = \max\left(\widetilde{v_{x_1}}\right)$ and $\widetilde{v_{j_1}} = \min\left(\widetilde{v_{x_1}}\right)$ for $\mathbf{v_x}$ representing the vectorial data within $\mathcal{S}$. In conclusion, the multivariate MM we propose choses as the extrema of a multivariate set the two vectors with the minimum and maximum principal components, for the definition of the erosion and dilation operations. The ordering of the two extrema is accomplished similarly to the PMM pseudo-extrema ordering, relative to three pairs (for the particular case of color images) of global references. The results generated through this approach are presented in the next section.

5 Results, Comparison and Discussion

In this section we describe, from a qualitative point of view, the results obtained with the presented approaches, for several widely-used color images (Fig. 3, 4 and 5). The PMM was applied using $k = 1.5$. One may notice the false colors introduced by the PMM, while the other two approaches generate colors from the initial image color distribution. In general, one may see that the erosions favor the growth of the dark objects (e.g. in Fig. 3 the background dissapears, the lips and eyes are emphasized), while the dilations shrink them (e.g. in Fig. 3 the lips and eyes disappear, while the background grid texture becomes uniform). The same effects may also be noticed in Fig. 4 and 5. However, in the PMM case, these effects are somehow attenuated by the filtering generated due to the inclusion of the data distribution within each local processing step. In addition, for PMM an edge enhancement may be observed, which is not present within the results generated with the other methods. In fact, the approach which uses the lexicographical order within the PCA space and our optimized distance approach generate very similar results, the maximum mean error between the generated results being equal to 4.12, computed between the images (e) and (m) from Fig. 4.

In Fig. 6 we present a detail from the results. It may be observed that the global PCA ordering approach may distort the original shapes, depending on the orientation of the local distribution relative to the global distribution used for PCA. The PMM approach preserves the shapes and also the textured regions, but it introduces false colors and it



(a) f

(b) $(\delta - \varepsilon)(f)$, global PCA, 3×3 SE

(c) $(\delta - \varepsilon)(f)$, PMM, 3×3 SE

(d) $(\delta - \varepsilon)(f)$, new approach, 3×3 SE

(e) $\varepsilon(f)$, 11×11 SE

(f) $\varepsilon(f)$, 5×5 SE

(g) $\delta(f)$, 5×5 SE

(h) $\delta(f)$, 11×11 SE

(i) $\varepsilon(f)$, 11×11 SE

(j) $\varepsilon(f)$, 5×5 SE

(k) $\delta(f)$, 5×5 SE

(l) $\delta(f)$, 11×11 SE

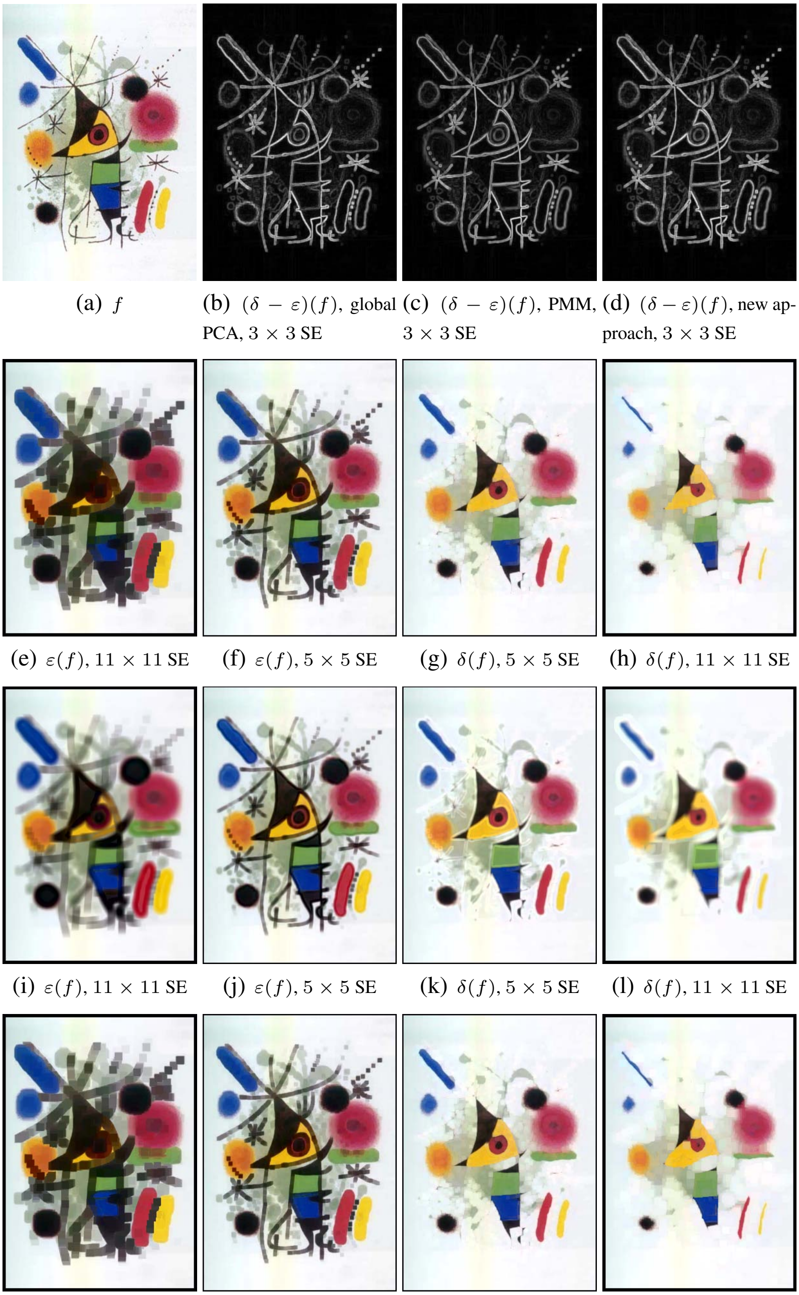
(m) $\varepsilon(f)$, 11×11 SE

(n) $\varepsilon(f)$, 5×5 SE

(o) $\delta(f)$, 5×5 SE

(p) $\delta(f)$, 11×11 SE

Fig. 3. Erosions, dilations and morphological gradients for global PCA lexicographical ordering (second row), PMM (third row) and the new approach – using optimal distance to probabilistically-estimated extrema (forth row).



(a) f

(b) $(\delta - \varepsilon)(f)$, global PCA, 3×3 SE

(c) $(\delta - \varepsilon)(f)$, PMM, 3×3 SE

(d) $(\delta - \varepsilon)(f)$, new approach, 3×3 SE

(e) $\varepsilon(f), 11 \times 11$ SE

(f) $\varepsilon(f), 5 \times 5$ SE

(g) $\delta(f), 5 \times 5$ SE

(h) $\delta(f), 11 \times 11$ SE

(i) $\varepsilon(f), 11 \times 11$ SE

(j) $\varepsilon(f), 5 \times 5$ SE

(k) $\delta(f), 5 \times 5$ SE

(l) $\delta(f), 11 \times 11$ SE

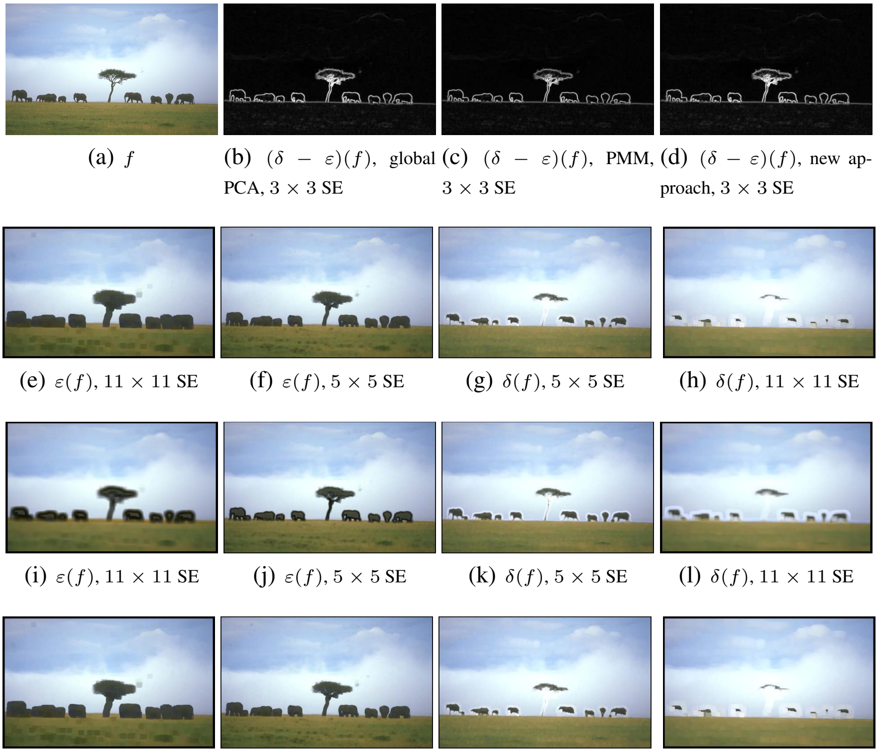
(m) $\varepsilon(f), 11 \times 11$ SE

(n) $\varepsilon(f), 5 \times 5$ SE

(o) $\delta(f), 5 \times 5$ SE

(p) $\delta(f), 11 \times 11$ SE

Fig. 4. Erosions, dilations and morphological gradients for global PCA lexicographical ordering (second row), PMM (third row) and the new approach – using optimal distance to probabilistically-estimated extrema (forth row).



(a) f

(b) $(\delta - \varepsilon)(f)$, global PCA, 3×3 SE

(c) $(\delta - \varepsilon)(f)$, PMM, 3×3 SE

(d) $(\delta - \varepsilon)(f)$, new approach, 3×3 SE

(e) $\varepsilon(f), 11 \times 11$ SE

(f) $\varepsilon(f), 5 \times 5$ SE

(g) $\delta(f), 5 \times 5$ SE

(h) $\delta(f), 11 \times 11$ SE

(i) $\varepsilon(f), 11 \times 11$ SE

(j) $\varepsilon(f), 5 \times 5$ SE

(k) $\delta(f), 5 \times 5$ SE

(l) $\delta(f), 11 \times 11$ SE

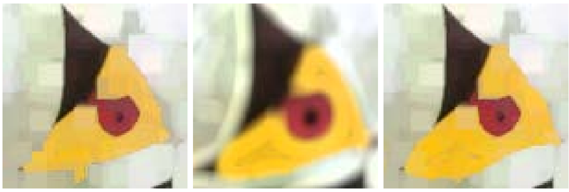
(m) $\varepsilon(f), 11 \times 11$ SE

(n) $\varepsilon(f), 5 \times 5$ SE

(o) $\delta(f), 5 \times 5$ SE

(p) $\delta(f), 11 \times 11$ SE

Fig. 5. Erosions, dilations and morphological gradients for global PCA lexicographical ordering (second row), PMM (third row) and the new approach – using optimal distance to probabilistically-estimated extrema (forth row).



(a) $\delta(f), 3 \times 3$ SE

(b) $\delta(f), 3 \times 3$ SE

(c) $\delta(f), 3 \times 3$ SE

Fig. 6. Detail from the results generated through the three described approaches: global PCA lexicographical ordering (left column), PMM (middle column) and our optimized distance approach (right column).

enhances the areas in the proximity of borders. The last MM approach solves the false color problem and also preserves the shapes better than the lexicographical method.

6 Conclusions

We started this paper by introducing a multivariate morphology based on a lexicographical order in the space obtained after using an orthogonal transformation. The orthogonal transformation is used in order to determine the most appropriate representation of correlated multivariate data and the axis with the maximum variance, based on the hypothesis that the extrema of a set are appropriately chosen in this axis system. We use this total order and the consequent morphology as a reference for comparison and discussion and we present the results obtained on color images, after applying a principal component analysis on the original image acquisition color space, i.e. RGB. The next step is to use the Tchebychev inequality in order to estimate in a probabilistic way the extrema of the multivariate data, along the first principal component of the local data. The *pseudo-infimum* and the *pseudo-supremum* constructed symmetrically around the mean allow to define morphological dilation and erosion. However, the estimated extrema may not belong to the original set, therefore false colors may appear in a color image application. In order to overcome this situation, we propose a new morphological approach, by choosing the points in the original set which are the closest to the probabilistic pseudo-extrema. The criteria we use for this choice is the maximization of the distance between the two extrema and the minimization of the distance between the probabilistic extrema estimated through the Tchebychev inequality and the data points of the original set. We proposed an algorithmic solution for the optimization problem. We compared all these three approaches for the case of color images represented in the RGB color set. We commented on the behavior of the erosions and dilations obtained. However, looking into detail we emphasized the differences in performance from a human perception point of view – the PMM is able to better preserve the texture of the object in the image, along the scales of the morphological operations. The lexicographical order and the new approach exhibit more non-linear behaviors and the results are very similar from a human perception point of view, thus showing that our morphological operators may be used in practical applications. In addition, through the approach presented in this framework, the infimum and the supremum can be conveniently chosen based on the pseudo-extrema given by the Tchebychev inequality and perceptual distance optimization.

Acknowledgement. This paper is partially supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), ID76945 financed from the European Social Fund and by the Romanian Government.

References

1. Matheron, G.: Random sets and integral geometry. John Wiley and Sons, Inc. (1975)
2. Serra, J.: Image Analysis and Mathematical Morphology, vol. I. Academic Press (1982)

3. Serra, J.: Image Analysis and Mathematical Morphology: Theoretical advances, vol. II. Academic Press Inc. (November 1987)
4. Sternberg, S.: Grayscale morphology. Computer Vision, Graphics, and Image Processing 35(3), 333–355 (1986)
5. Maragos, P.: A representation theory for morphological image and signal processing. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(6), 586–599 (1989)
6. Maragos, P., Schafer, R.: A unification of linear, median, order-statistics and morphological filters under mathematical morphology. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1985, vol. 10, pp. 1329–1332 (April 1985)
7. Meyer, F., Beucher, S.: The morphological approach of segmentation: the watershed transformation. In: Dougherty, E. (ed.) Mathematical Morphology in Image Processing. Marcel Dekker, New York (1992)
8. Maragos, P.: Pattern spectrum and multiscale shape representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(7), 701–716 (1989)
9. Ronse, C.: Why mathematical morphology needs complete lattices. Signal Processing 21(2), 129–154 (1990)
10. Barnett, V.: The ordering of multivariate data. Journal of the Royal Statistical Society 139(3), 318–355 (1976)
11. 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)
12. Aptoula, E., Lefèvre, S.: A comparative study on multivariate mathematical morphology. Pattern Recognition 40(11), 2914–2929 (2007)
13. Velasco-Forero, S., Angulo, J.: Mathematical morphology for vector images using statistical depth. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 355–366. Springer, Heidelberg (2011)
14. Velasco-Forero, S., Angulo, J.: Random projection depth for multivariate mathematical morphology. IEEE Journal of Selected Topics in Signal Processing 6(7), 753–763 (2012)
15. 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)
16. Aptoula, E., Lefèvre, S.: α-trimmed lexicographical extrema for pseudo-morphological image analysis. Journal of Visual Communication and Image Representation 19(3), 165–174 (2008)
17. Ivanovici, M., Caliman, A., Richard, N., Fernandez-Maloigne, C.: Towards a multivariate probabilistic morphology for colour images. In: Proceedings of the 6th European Conference on Colour in Graphics, Imaging and Vision, Amsterdam, the Netherlands, May 6-9, pp. 189–193 (2012)
18. Goutsias, J., Heijmans, H.J.A.M., Sivakumar, K.: Morphological operators for image sequences. Comput. Vis. Image Underst. 62(3), 326–346 (1995)
19. Angulo, J.: Morphological colour operators in totally ordered lattices based on distances: Application to image filtering, enhancement and analysis. Computer Vision and Image Understanding 107(1-2), 56–73 (2007)
20. Soille, P.: Morphological Image Analysis: Principles and Applications. Springer (2002)
21. Papoulis, A.: Probability, random variables, and stochastic processes, 3rd edn. McGraw-Hill (1991)
22. Caliman, A., Ivanovici, M., Richard, N.: Probabilistic mathematical morphology for greyscale, color and multivariate images. Submitted to Pattern Recognition

Group-Invariant Frames for Colour Morphology*

Jasper J. van de Gronde** and 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.j.van.de.gronde@rug.nl

Abstract. In theory, there is no problem generalizing morphological operators to colour images. In practice, it has proved quite tricky to define a generalization that "makes sense". This could be because many generalizations violate our implicit assumptions about what kind of transformations should not matter. Or in other words, to what transformations operators should be invariant. As a possible solution, we propose using frames to explicitly construct operators invariant to a given group of transformations. We show how to create saturation- and rotation-invariant frames, and demonstrate how group-invariant frames can improve results.

Keywords: colour morphology, group invariance, frames.

1 Introduction

Mathematical morphology is based on being able to order images, and specifically on being able to compute their supremum and infimum. For binary images the order is straightforward, just use set inclusion. For greyscale images it is similarly straightforward. For colour images the problem is considerably more difficult though. Not in a theoretical sense, as various schemes can be (and have been) used to define orders on colour values (and by extension on colour images). The problem is in making the theory line up with our (implicit) expectations.

Talbot et al. [10] produced one of the earlier publications in mathematical morphology that identified this problem. Since then, many authors have tried various approaches around this problem, which has been called the "false colour problem" (see, for example, Serra's paper [9] by this name). The name derives from the appearance of new ("false") colours that were not present in the original image, as illustrated in Fig. 1. This is because the most basic (and sensible) ordering compares colours per-channel, resulting in a per-channel supremum/infimum.

One obvious way out is to somehow impose a total order on the colour space. However, in general it really does not make sense to enforce that red and green must come in some order. The most interesting work in this direction is probably statistics based, like the approach described by Velasco-Forero and Angulo [11]

* This research is funded by the Dutch National Science Foundation (NWO), project no. 612.001.001.

** Corresponding author.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 267–278, 2013.

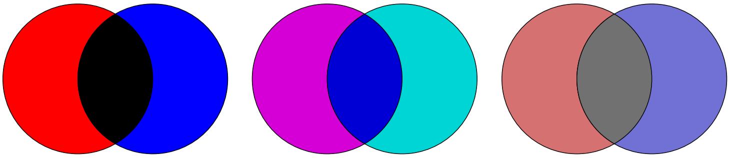


Fig. 1. The infimum of blue and red is black (in the RGB colour space). In contrast, the infimum of cyan and magenta is blue, and if we desaturate red and blue, then their infimum suddenly becomes grey. To an observer (who does not know about RGB colour spaces), it is not immediately obvious why this should be so. Why is the infimum of blue and red not also some in-between (purplish) colour for example?

that orders colours according to their statistical depth in a particular image. In any case, enforcing a total order on a multi-dimensional space necessarily introduces discontinuities in the supremum/infimum (in the sense that we could have $a < b < a'$ and $\|a - b\| \gg \|a - a'\|$). There is thus still a need for a general-purpose framework for multivariate morphology.

In designing such a general-purpose framework for morphology on colour (and other multivariate) images, is it really necessary to avoid introducing *any* new colours? Linear filters do it all the time, and even a traditional structural greyscale dilation with a non-flat structuring element can introduce new values. Also, in tests, the per-channel approach often works quite well, as demonstrated by Aptoula and Lefèvre [2]. Hence, we believe the problem with "false colours" is not that they are new, but rather that they are unexpected or unintuitive.

Goutsias et al. [5] have argued that the problem with handling channels independently is that it ignores the correlation between channels. Similarly, Astola et al. [3] show how a yellow pixel near a boundary between a green and a red region is not removed by a median filter, but just moved. In principle, this is a well-known problem with any kind of median filter. But now that it happens with a colour image, it is worse. We perceive the colour yellow to be something different from either red or green, while a computer (using an RGB colour space) considers it to be a combination of red and green. So although humans treat yellow on equal footing with red and green[1], the computer does not.

We believe the false colour problem might stem from violating certain implicit assumptions. In particular, as humans, we typically do not think of vectors in terms of their components in a specific basis. As Serra [8] put it: "..., there exists an infinite number of other equivalent systems of coordinates for the same vector space; they derive from the first one by rotations, similarities, passages to spherical, cylindric or polar coordinates, etc." In this spirit, our solution is to formalize our assumptions as a group of transformations that lead to equivalent systems of coordinates, and to develop a method for making arbitrary operators invariant to those transformations. This builds upon our earlier work [6], providing a much more compact statement of the main result and enlarging its scope to include certain non-orthogonal transforms (saturation scaling). Also, we now

[1] See [7] for several interesting expositions on colour perception, especially chapter four is highly relevant in the current context.

provide a qualitative evaluation of the effects of our method, illustrating why and when invariance leads to better results.

2 Definitions

A *Hilbert space* H is a vector space with an inner product (which is also complete w.r.t. the metric induced by the inner product, but this is of little interest here). The inner product is denoted by '$\cdot$' and should be positive definite. The norm of a vector $\mathbf{a} \in H$ is defined as $\|\mathbf{a}\| = \mathbf{a} \cdot \mathbf{a}$. If $\{\mathbf{e}_k\}_{k\in\mathcal{K}}$ (with $\mathcal{K}$ a finite index set) is a basis for H, then there also exists a dual basis $\{\mathbf{e}^k\}_{k\in\mathcal{K}}$, such that for any $\mathbf{a} \in H$: $\mathbf{a} = \sum_{k\in\mathcal{K}}(\mathbf{e}^k \cdot \mathbf{a})\,\mathbf{e}_k$. Since we will only be working with Hilbert spaces on the reals, a Hilbert space with a basis $\{\mathbf{e}_k\}_{k\in\mathcal{K}}$ will be denoted by $\mathbb{R}^{\mathcal{K}}$. Note the use of bold face letters for vectors.

Instead of using a basis to span a Hilbert space $\mathbb{R}^{\mathcal{K}}$, we can also use a *frame* [4]. A frame is a set of vectors $\{\mathbf{f}_i\}_{i\in\mathcal{I}}$ (not necessarily finite or even countable) such that there are finite positive constants A and B for which (for all $\mathbf{a} \in \mathbb{R}^{\mathcal{K}}$)

$$A\,\|\mathbf{a}\|^2 \leq \|F\mathbf{a}\|^2 \leq B\,\|\mathbf{a}\|^2. \tag{1}$$

Here F is the *analysis operator* of the frame, defined by $(F\mathbf{a})_i = \mathbf{f}_i \cdot \mathbf{a}$. For simplicity, assume that the range of the analysis operator is again a Hilbert space $\mathbb{R}^{\mathcal{I}}$. This is generally true for the cases we are interested in.

Like a basis, a frame has a dual frame. One of the interesting properties of a frame, however, is that there need not be just one dual frame; typically there are infinitely many dual frames. All dual frames of a frame have an associated *synthesis operator* which acts as a (left-)inverse of the analysis operator for the frame. Here we will mainly concern ourselves with the *canonical* dual frame. This particular choice gives the least-squares solution $\mathbf{a}$ of $F\mathbf{a} = \mathbf{u}$. For this reason we will denote the synthesis operator associated with the canonical dual frame by F^+ (to evoke associations with the Moore-Penrose pseudoinverse).

It is important to note that if A equals B in Eq. (1), a frame is called *tight*, and that its canonical dual is the frame itself, multiplied by $\frac{1}{A}$. The corresponding synthesis operator is then the *adjoint* F^* of the analysis operator F, multiplied by $\frac{1}{A}$. The adjoint is defined by $\mathbf{u} \cdot (F\mathbf{a}) = (F^*\mathbf{u}) \cdot \mathbf{a}$ for all $\mathbf{a} \in \mathbb{R}^{\mathcal{K}}$ and $\mathbf{u} \in \mathbb{R}^{\mathcal{I}}$. For (real) matrices, the adjoint is simply the transpose. In summary, if a frame is tight, then $F^+ = \frac{1}{A}F^*$.

A transformation group $\mathbb{T}$ on a Hilbert space $\mathbb{R}^{\mathcal{K}}$ is a set of invertible mappings of $\mathbb{R}^{\mathcal{K}}$ onto itself, with an associative binary operation '$\circ$' (function composition). As a group it must be closed for composition, it must contain the identity mapping, and it must contain the inverse of every transformation in the group. An operator $\phi : \mathbb{R}^{\mathcal{K}} \to \mathbb{R}^{\mathcal{K}}$ is invariant to $\mathbb{T}$ if it commutes with all transformations in $\mathbb{T}$, so $\forall\tau \in \mathbb{T}(\phi \circ \tau = \tau \circ \phi)$.

3 Construction

Let us assume that $\mathbb{T}$ is a group of linear transformations on the Hilbert space $\mathbb{R}^{\mathcal{K}}$, and that ϕ_0 is an operator on $\mathbb{R}^{\mathcal{K}}$ that is not invariant to $\mathbb{T}$. If $\{\mathbf{e}_k\}_{k\in\mathcal{K}}$ is the original basis used for $\mathbb{R}^{\mathcal{K}}$, then we consider the set $\{\mathbf{f}_i\}_{i\in\mathcal{I}}$, with $\mathcal{I} = \mathbb{T}\times\mathcal{K}$ and $\mathbf{f}_{\tau,k} = \tau^*\,\mathbf{e}^k$ (where τ^* is the adjoint of τ). Under suitable conditions this set forms a frame that can be used to construct a $\mathbb{T}$-invariant version of ϕ_0.

Note that the above choice of frame has the interesting property that the analysis operator essentially corresponds to taking the components of all transformed versions of a vector:

$$(F\mathbf{a})_{\tau,k} = (\tau^*\,\mathbf{e}^k)\cdot\mathbf{a} = \mathbf{e}^k\cdot(\tau\,\mathbf{a}) = (\tau\,\mathbf{a})_k.$$

As a consequence, $(F\mathbf{a})_\tau$ is considered equal to $\tau\,\mathbf{a}$. Similarly, if $\mathbf{u}\in\mathbb{R}^{\mathcal{I}}$, then $\mathbf{u}_\tau$ (with $\tau\in\mathbb{T}$) denotes the vector in $\mathbb{R}^{\mathcal{K}}$ described by the coefficients $\{u_{\tau,k}\}_{k\in\mathcal{K}}$.

To construct an operator $\phi:\mathbb{R}^{\mathcal{K}}\to\mathbb{R}^{\mathcal{K}}$ that is invariant to $\mathbb{T}$, based on an operator ϕ_0, we first define an operator $\psi:\mathbb{R}^{\mathcal{I}}\to\mathbb{R}^{\mathcal{I}}$ such that $\psi(\mathbf{u})_\tau = \phi_0(\mathbf{u}_\tau)$ for all $\tau\in\mathbb{T}$ and $\mathbf{u}\in\mathbb{R}^{\mathcal{I}}$. So $\psi(F\mathbf{a})$ computes $\phi_0(\tau\,\mathbf{a})$ for all transformations $\tau\in\mathbb{T}$. We then simply define ϕ as $F^+\circ\psi\circ F$, which Theorem 1 shows to be invariant to $\mathbb{T}$ under a mild condition on the norm on $\mathbb{R}^{\mathcal{I}}$.

Theorem 1. *Assume ϕ_0 is an operator on $\mathbb{R}^{\mathcal{K}}$, and that $\psi:\mathbb{R}^{\mathcal{I}}\to\mathbb{R}^{\mathcal{I}}$ is defined by $\psi(\mathbf{u})_\tau = \phi_0(\mathbf{u}_\tau)$ (for all $\tau\in\mathbb{T}$ and $\mathbf{u}\in\mathbb{R}^{\mathcal{I}}$). The operator $\phi:\mathbb{R}^{\mathcal{K}}\to\mathbb{R}^{\mathcal{K}}$, defined as $\phi = F^+\circ\psi\circ F$, is then invariant to the transformation group $\mathbb{T}$ on the Hilbert space $\mathbb{R}^{\mathcal{K}}$, provided the norm on $\mathbb{R}^{\mathcal{I}}$ is invariant to a permutation of $\mathbb{T}$ (in the sense that if P is a permutation operator mapping each index $(\tau,k)\in\mathcal{I}$ to some index $(p(\tau),k)\in\mathcal{I}$, then $\|\mathbf{u}\| = \|P\mathbf{u}\|$ for any choice of $\mathbf{u}\in\mathbb{R}^{\mathcal{I}}$)*[2].

Proof. It should be clear that $(F\tau\,\mathbf{a})_\sigma = \sigma\,\tau\,\mathbf{a} = (F\mathbf{a})_{\sigma\tau}$ for all $\tau,\sigma\in\mathbb{T}$ and $\mathbf{a}\in\mathbb{R}^{\mathcal{K}}$. Thus, $\psi(F\tau\,\mathbf{a})_\sigma = \psi(F\mathbf{a})_{\sigma\tau}$. In other words, $\psi(F\tau\,\mathbf{a})$ is a permuted version of $\psi(F\mathbf{a})$. Denote the effect of this permutation by the operator P, so that $F\circ\tau = P\circ F$ and $\psi(F\tau\,\mathbf{a}) = P\,\psi(F\mathbf{a})$. Similarly, P^{-1} is used to denote the inverse permutation, and we naturally have $F\circ\tau^{-1} = P^{-1}\circ F$. What remains, is to show that $F^+\circ P = \tau\circ F^+$.

By definition, $F^+\mathbf{u}$ (with $\mathbf{u}\in\mathbb{R}^{\mathcal{I}}$) is the least-squares solution $\mathbf{a}$ to $F\mathbf{a} = \mathbf{u}$, while $F^+P\mathbf{u}$ is the least-squares solution $\mathbf{b}$ to $F\mathbf{b} = P\mathbf{u}$. As these linear systems are over-determined, these least-squares solutions are unique. It is thus sufficient to show that $\mathbf{b}$ must equal $\tau\,\mathbf{a}$. This follows directly from the invariance of the norm to any permutation. Due to this, the least-squares solution to $F\mathbf{b} = P\mathbf{u}$ must equal the least-squares solution to $(P^{-1}\circ F)\mathbf{b} = \mathbf{u}$, or equivalently, $F\tau^{-1}\,\mathbf{b} = \mathbf{u}$. We can now see that $\mathbf{b}$ must indeed equal $\tau\,\mathbf{a}$, and thus $F^+\circ P$ must equal $\tau\circ F^+$. We thus have $\phi(\tau\,\mathbf{a}) = F^+\,\psi(F\tau\,\mathbf{a}) = F^+P\,\psi(F\mathbf{a}) = \tau\,F^+\,\psi(F\mathbf{a}) = \tau\,\phi(\mathbf{a})$. This concludes the proof.

3.1 Rotation Invariance

Now we will show how to construct operators on RGB colour images that are invariant to all rotations of the colour space. For simplicity, the RGB colour space is considered to be $\mathbb{R}^3$ ($\mathbb{R}^{\mathcal{K}}$ with $\mathcal{K} = \{1,2,3\}$), with an orthonormal basis

[2] This condition is connected to the concept of a “Haar measure”.

corresponding to the red, green and blue components. The group $\mathbb{T}$ of *all* 3D rotations is $SO(3)$. The image/range of the frame analysis operator can thus be considered to be $\mathbb{R}^{SO(3)\times 3}$. Elements of $\mathbb{R}^{SO(3)\times 3}$ should be interpreted as vectors whose components can be indexed by elements from $\mathcal{I} = SO(3) \times \{1, 2, 3\}$.

The next task is to define a suitable inner product on $\mathbb{R}^{SO(3)\times 3}$. We will build our inner product on top of the one for $\mathbb{R}^3$. As we saw in Theorem 1, the inner product should be invariant with respect to permutations of $SO(3)$. Take $\int_{SO(3)} f(r)\,\mathrm{d}r$ to be the integral of $f(r)$ over the entire group of rotations (the rotations represented by r), weighing all rotations "equally"[3]. We assume that $\int_{SO(3)} 1\,\mathrm{d}r$ is some finite non-zero constant A. The inner product can then simply be defined as follows for $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^{SO(3)\times 3}$:

$$\mathbf{u} \cdot \mathbf{v} = \int_{SO(3)} \mathbf{u}_r \cdot \mathbf{v}_r \,\mathrm{d}r.$$

It can be verified that this inner product is invariant to the permutations alluded to in Theorem 1 (which makes sense, given that we weigh all rotations equally).

Now we need to find F^+. This proves particularly easy, since the frame is tight (A equals B in Eq. (1), see the second to last paragraph in Eq. (1)):

$$\|F\mathbf{a}\|^2 = (F\mathbf{a}) \cdot (F\mathbf{a}) = \int_{SO(3)} \|(F\mathbf{a})_r\|^2 \,\mathrm{d}r = \int_{SO(3)} \|r\,\mathbf{a}\|^2 \,\mathrm{d}r = A\,\|\mathbf{a}\|^2.$$

The last step is valid because rotations are orthogonal. In conclusion: $F^+ = \frac{1}{A}F^*$.

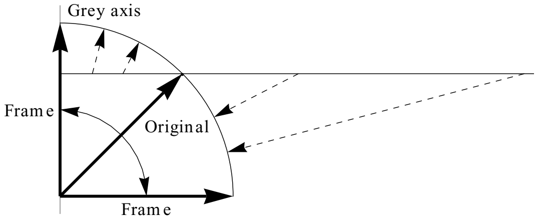
In practice, we can use a finite set of vectors to approximate the frame of all rotations of the original basis vectors. One method is to sample the group of all rotations, and construct a frame (and canonical dual) based on this. Alternatively, we can also take a sample of uniformly distributed unit vectors, and construct a discrete frame based on these, with analysis operator $\hat{F}$. We can then take the Moore-Penrose pseudoinverse of $\hat{F}$ to find the canonical dual frame.

Note that the above is geared towards single colour values, rather than colour images. For treating colour images we simply apply the same technique per-pixel. So to filter a colour image using the rotation-invariant frame discussed here, we would first compute a "greyscale" image for every vector in the frame, then apply some greyscale morphological operator on each of these images, and then finally combine all the greyscale images (according to the canonical dual frame).

3.2 Saturation Invariance

As shown in our earlier work [6], the construction used above breaks down when the transformations have eigenvalues with non-unit magnitude. However, in some cases, we can still construct frames invariant to such transformations. We will illustrate this by constructing a frame invariant to scaling the saturation of a colour in the RGB colour space. The saturation, or colourfulness, is taken to be the distance to the grey axis, the line through both white and black.

[3] Formally, we take the Haar integral.



Original $\mathbf{e}^1 = (1,0)$, grey vector $\mathbf{g} = (1,1)$, $S_s\,\mathbf{e}^1 = s\,(1,0) + (1-s)\,\frac{1}{2}(1,1) = \frac{1}{2}(1+s, 1-s)$, frame:

$$\mathbf{f}_{0,1} = \lim_{s\downarrow 0} \frac{(1+s,1-s)}{\sqrt{2(1+s)^2}} = \frac{1}{\sqrt{2}}(1,1),$$

$$\mathbf{f}_{\infty,1} = \lim_{s\uparrow\infty} \frac{(1+s,1-s)}{\sqrt{2(1+s)^2}} = \lim_{s\uparrow\infty} \frac{(s,-s)}{\sqrt{2\,s^2}} = \frac{1}{\sqrt{2}}(1,-1).$$

Fig. 2. Illustration of the process for constructing a saturation-invariant frame. If the original basis vector is scaled perpendicularly to the grey axis, it traces the horizontal line. If the operator ϕ_0 is invariant to scalar multiplication, then we can avoid vectors of arbitrarily large magnitude by normalizing them. This maps the horizontal line (of infinite extent) onto the quarter circle. For the frame we only take the limit vectors on either end; these are eigenvectors of all scalings.

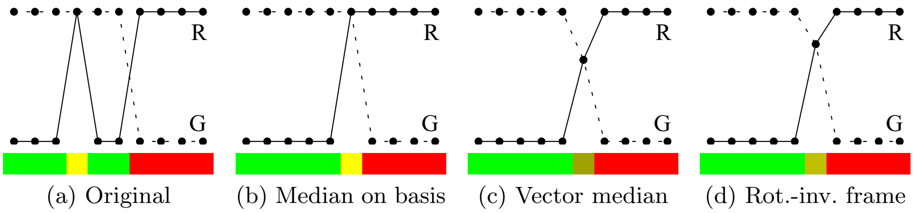
As group $\mathbb{T}$ we take the set of linear transformations $\{S_s \mid s \in \mathbb{R} \text{ and } 0 < s\}$, with S_s defined by $S_s\,\mathbf{a} = s\,\mathbf{a} + (1-s)\,\frac{\mathbf{a}\cdot\mathbf{g}}{\mathbf{g}\cdot\mathbf{g}}\mathbf{g}$ for $\mathbf{a} \in \mathbb{R}^3$ and $\mathbf{g}$ representing grey (it is not particularly important which shade of grey). These transformations only scale the component of $\mathbf{a}$ that is perpendicular to the grey axis, and can thus be interpreted as scaling the saturation.

The next step is to assume that the operator ϕ_0 is already invariant to a uniform scaling of the colour space (so scaling all components equally). This is typically the case for morphological operators, and allows us to normalize all vectors. Now, instead of creating a frame indexed by $\mathbb{T} \times \{1,2,3\}$, we create a frame indexed by $\mathcal{I} = \{0,\infty\} \times \{1,2,3\}$ (as illustrated in Fig. 2). Noting that $S_s^* = S_s$ for all $s \in \mathbb{R}$, the six frame vectors $\{\mathbf{f}_i\}_{i\in\mathcal{I}}$ are then defined as

$$\mathbf{f}_{0,k} = \lim_{s\downarrow 0} \frac{S_s\,\mathbf{e}^k}{\|S_s\,\mathbf{e}^k\|} \quad \text{and} \quad \mathbf{f}_{\infty,k} = \lim_{s\uparrow\infty} \frac{S_s\,\mathbf{e}^k}{\|S_s\,\mathbf{e}^k\|}.$$

In practice, this means we get a frame consisting of a grey vector and three vectors that are perpendicular to that grey vector. These vectors are all eigenvectors of *all* $S_s \in \mathbb{T}$. In particular, for some $S_s \in \mathbb{T}$, the grey vector ($\mathbf{f}_{0,1} = \mathbf{f}_{0,2} = \mathbf{f}_{0,3}$) has eigenvalue 1, while the other three vectors ($\mathbf{f}_{\infty,1}$, $\mathbf{f}_{\infty,2}$ and $\mathbf{f}_{\infty,3}$) have eigenvalue s. Recalling that $S_s = S_s^*$, this means that $(FS_s\mathbf{a})_0 = (F\mathbf{a})_0$ and $(FS_s\mathbf{a})_\infty = s\,(F\mathbf{a})_\infty$ for any $S_s \in \mathbb{T}$. Since we assumed that ϕ_0 is invariant to multiplication by a scalar, we can easily show that $F^+ \circ \psi \circ F$ is invariant to $\mathbb{T}$, with ψ as in Theorem 1.

It should be noted that the above approach is not terribly useful for much more general scalings. For example, illumination changes are often modelled by multiplying the red, green and blue channels with independent weights. The eigenvectors of such a transformation are in general obviously only the red, green and blue vectors, thus the only possible "frame" would be the original basis. So a different method would be needed to combine illumination changes and other (non-linear) transformations, discussed by Angulo [1, §3.4], with rotations.



(a) Original (b) Median on basis (c) Vector median (d) Rot.-inv. frame

Fig. 3. The original (a) is filtered using a median filter applied independently to the channels using the original basis (b), a vector median filter (c), and a median filtered applied independently to the "channels" in a rotation-invariant frame (d). In each case, both a plot of the red (R) and green (G) channels, as well as the actual (colour) image are shown. The spike in the red channel in (a) is assumed to be noise. It gives rise to a yellow band between the red and green regions in (b). In contrast, using the vector median filter or the rotation-invariant frame, it is much clearer that there is a transition between red and green (note, though, that the frame approach is much more general).

4 Results

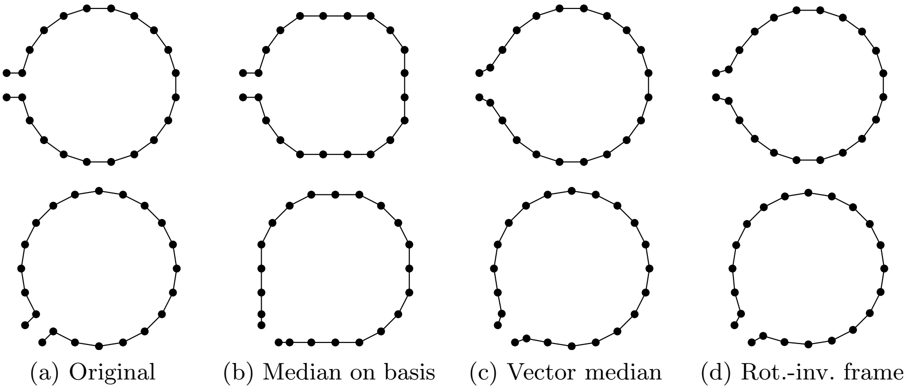
We briefly show some of the results that can be attained using group-invariant morphology. To show how it compares to more traditional solutions, we start by examining two examples given by Astola et al. [3]. In the first example (Fig. 3), the per-channel median filter does not pick up on the fact that yellow is a different colour from both red and green, and thus finds that just before the switch to red, there are three out of five pixels that have a non-zero red component, instead of "seeing" one yellow pixel, two green pixels and two red pixels. In contrast, the other two filters indeed give much less importance to the yellow pixel, resulting in a more natural transition from red to green.

The second example by Astola et al. (see Fig. 4) shows how processing channels independently can result in an unnatural bias towards filtering only along the axes. If we have a vector with completely unrelated components, like average temperature and population density, then this might be fine. But here it is just as meaningful to choose a different basis, so we would not expect a filter to show a bias towards filtering in specific directions.

Astola et al. [3] suggested solving both these issues by creating what they called a "vector median filter". This was based on minimizing the sum of the distances to all vectors. As can be seen in Figs. 3 and 4, our method gives very similar results[4]. However, in contrast to the vector median filter, our approach generalizes easily to *any* operator (not just the median filter), and simply follows logically from enforcing certain constraints.

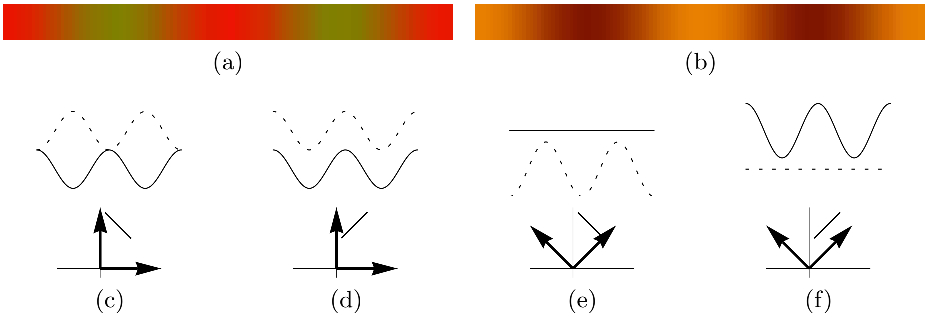
Another example of why it can be useful to have rotation invariance, is given in Fig. 5. Looking at the channels independently, both signals have an oscillation

[4] It should be noted that the original vector median filter always picked the result from one of the input values. We have chosen not to do this, as this forces an arbitrary decision to be made in ambiguous cases (like the one shown in Fig. 3).



(a) Original (b) Median on basis (c) Vector median (d) Rot.-inv. frame

Fig. 4. The original (a) is filtered using a median filter applied independently to the channels in the original basis (b), a vector median filter (c), and a median filter applied to the "channels" in a rotation-invariant frame (d). The 1D signals are plotted as strings of points in the 2D value(/colour) space. In (b) there is a clear flattening of the result in the direction of the axes, and a simple 45° rotation of the input results in a completely different output. In contrast, neither the vector median filter, nor the median filter using the rotation-invariant frame, shows such a directional bias.



(a) (b)

(c) (d) (e) (f)

Fig. 5. The two signals in (a) and (b) are plotted using the original basis in (c) and (d), respectively. In (c) to (f), the signals are plotted against "time" (top) and in the 2D value space (bottom), like in Fig. 4. The basis vectors used for the middle row are shown as arrows in the bottom row. In the original basis the two signals are indistinguishable if one just looks at the frequency and amplitude of the oscillations in the channels independently. Using a different basis, like in (e) and (f), the signals are clearly different though. In a rotation-invariant frame we simply use all possible bases, so we always pick up on such differences.

in both channels, all at the same frequency. However, there *is* a really clear difference, if one considers oscillations in other directions as well. Previously, a rotation-invariant frame was indeed shown [6] to lead to better performance in a texture classification task based on an autocorrelation-like operator.

Figures 6 and 7 show some results on more natural images. Figure 6 shows the result of applying a dilation using the original basis, as well as using a

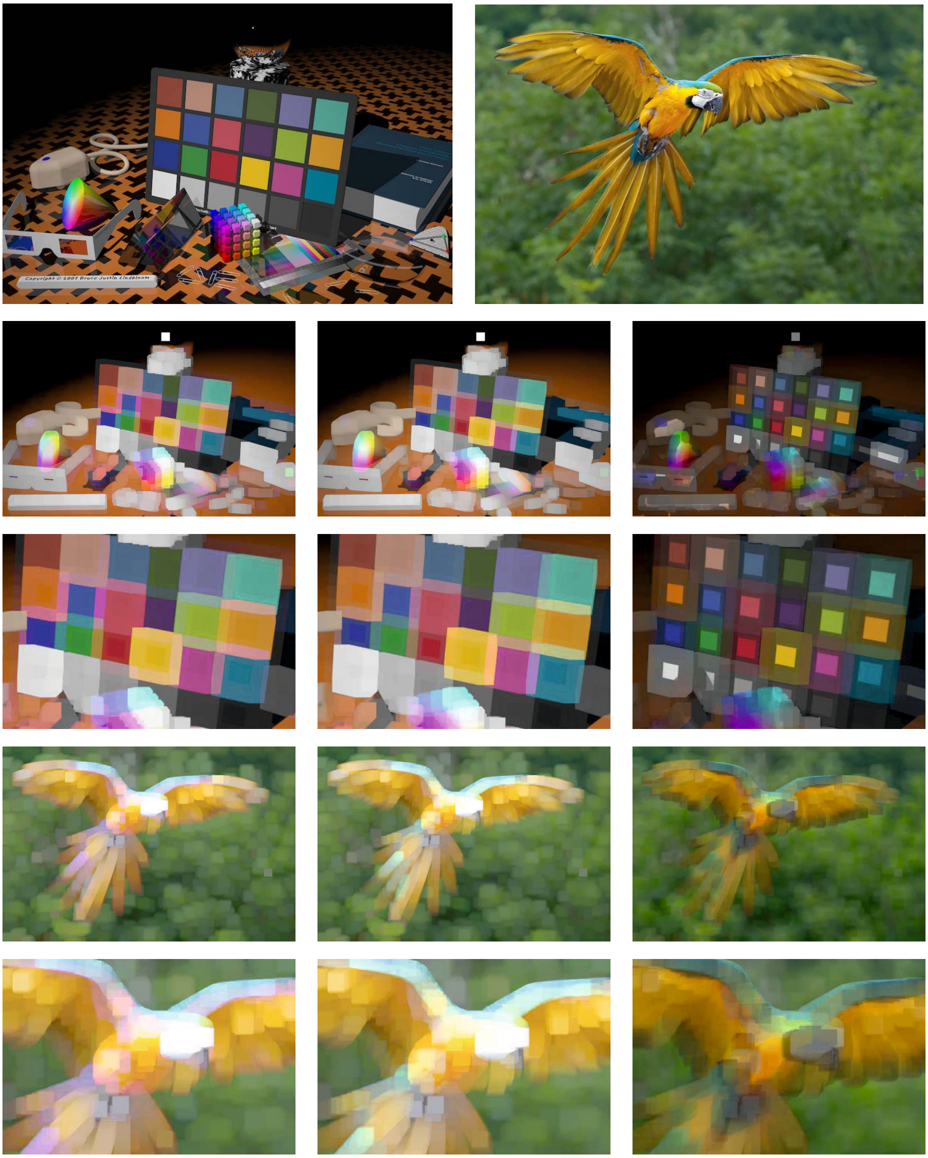


Fig. 6. Dilations (by a 21×21 square) of the originals (top row, both 768×512 pixels) using different frames, from left to right: the RGB basis, a hue-invariant frame and a rotation-invariant frame. Rows three and five are zoomed versions of rows two and four. A pink haze appears between orange and blue patches on the colour chart when using the RGB basis. The same kind of effect is quite visible all over the parrot. Using a hue-invariant frame solves these problems (middle column). Using the rotation-invariant frame (right column) is similar to averaging a dilation and an erosion, but otherwise shows no significant colour artefacts. (The parrot image is based on a photograph by Luc Viatour / `www.Lucnix.be`, used under the CC BY 2.0 license.)

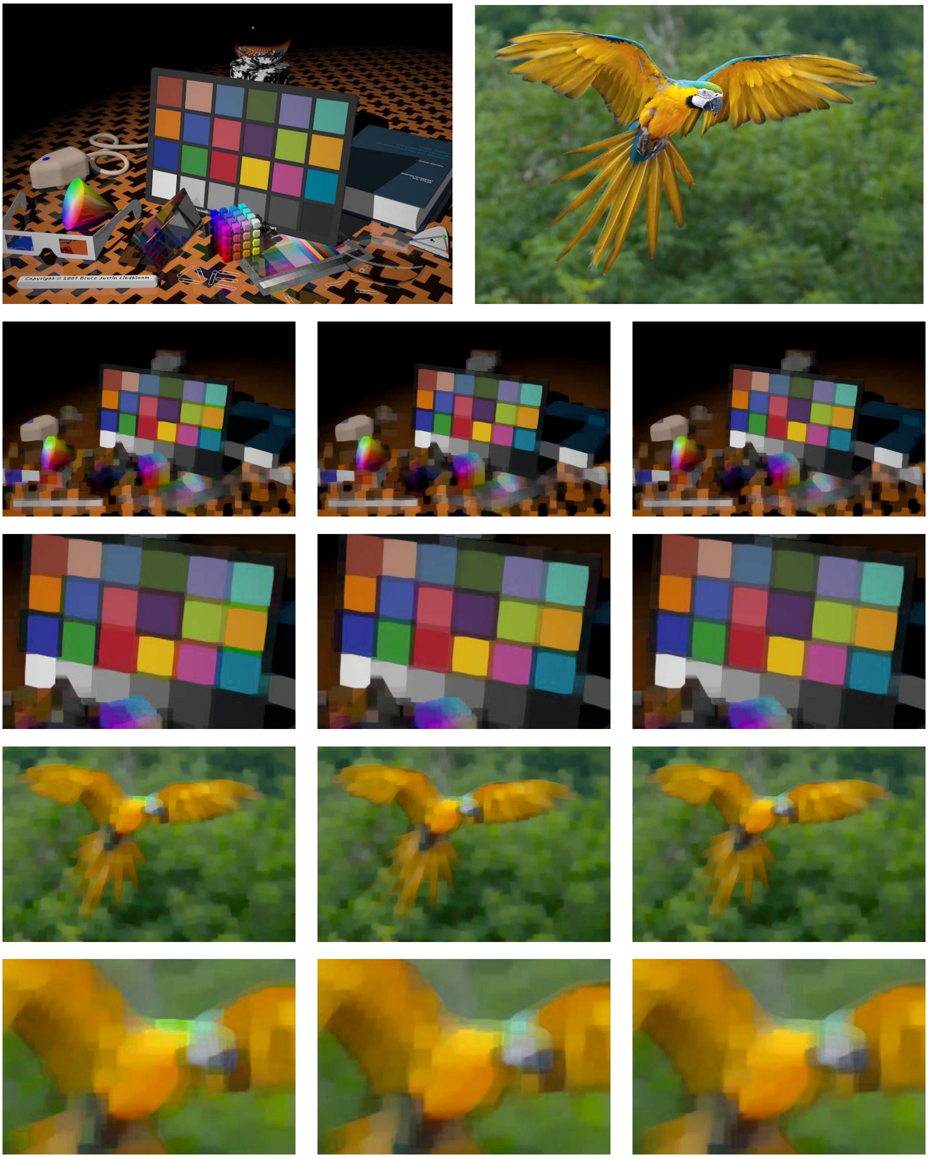


Fig. 7. Similar to Fig. 6, except that now the self-dual OCCO operator is used, with a 15×15 square structuring element. Again colour artefacts are visible in the left column, especially between the orange and blue patches on the colour chart (the transition region becomes green). Using a hue-invariant frame (middle column), and especially using a rotation-invariant frame (right column), eliminates these artefacts. Similarly, in the left column the back of the neck of the parrot is suddenly bright green, even though it is originally blueish, and there is only yellow and dark green in the neighbourhood. Again this artefact is largely gone in the middle and right columns.

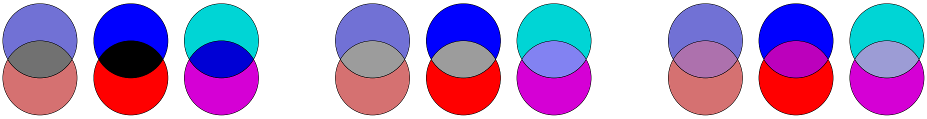


Fig. 8. Three different pairs of colours and their infima using different frames. From left to right: the original basis (same as in Fig. 1), a saturation-invariant frame and a saturation and hue-invariant frame. Gamma correction (sRGB) was used; this avoids big differences in lightness between the saturated and desaturated colours.

hue-invariant frame and a rotation-invariant frame. Here a hue-invariant frame is taken to be a frame that is generated using rotations around the grey axis (the axis running through all grey-ish colours, including black and white), rather than all rotations. This is explained in more detail in van de Gronde and Roerdink [6]. Figure 7 does the same for the OCCO operator. The OCCO operator is a self-dual operator[5] consisting of the average of an opening of a closing and a closing of an opening. Operators like this are excellent candidates for use with a rotation-invariant frame. In fact, the OCCO operator can be derived from taking the opening of a closing as ϕ_0 in Theorem 1, with $\mathbb{T}$ the group consisting of the identity mapping and the inversion mapping.

The effect of saturation invariance is illustrated in Fig. 8. Without saturation invariance, the infimum of pure red and blue is black, while the infimum of desaturated red and blue is grey. With a saturation-invariant frame, the infimum of pure red and blue is grey as well. Conceptually, this makes a lot more sense; both colours are pretty light. Using a hue *and* saturation-invariant frame provides an interesting alternative to filtering in the HSL colour space. Instead of representing saturation and hue using a magnitude and an angle (which is hard to order sensibly) we effectively represent them as a function of angle.

5 Conclusion

The false colour problem is the appearance of new colours that bear no obvious resemblance to the original colours, as a result of processing the colour channels independently. We suggest that the actual problem lies in violating certain invariances that we implicitly assume should hold. As a potential solution, we provide a method for modifying any given operator so that it becomes invariant to a given group of transformations. This essentially constitutes a much more compact statement of our previous result [6]. Furthermore, it is shown that this result can be extended to certain non-orthogonal transformations.

Using several basic examples we illustrate how our approach can lead to more intuitive and better quality results. A practical implementation is fairly straightforward: the original operator simply has to be called multiple times, on different, transformed, versions of the original. The main problem lies in the increased processing required. In the (2D) examples shown here the frames were already

[5] A self-dual operator is taken to be invariant to inverting the image.

about ten times as large as the original basis (which directly translates to ten times the processing time[6]). It would thus be worthwhile to look into methods for decreasing the amount of processing needed.

In future work, it would be interesting to examine alternatives for using the canonical dual frame to get back to the original colour space. The canonical dual frame leads to a least-squares solution and simple, linear, methods. However, other methods, based on different dual frames or L_p-minimization with $p \neq 2$ for example, might also have interesting characteristics.

Finally, we only looked at colour images, but there is absolutely no reason the same theory could not be applied to other kinds of vector-valued images. In particular, we will attempt to apply the same idea to diffusion tensor images. Other examples that could be interesting to look at of course include hyperspectral images, but also light fields for example. Essentially our approach might be interesting for any multivariate data where it makes sense to mix components.

References

[1] Angulo, J.: Geometric algebra colour image representations and derived total orderings for morphological operators Part I: Colour quaternions. J. Vis. Commun. Image R. 21(1), 33–48 (2010)

[2] Aptoula, E., Lefèvre, S.: A comparative study on multivariate mathematical morphology. Pattern Recognition 40(11), 2914–2929 (2007)

[3] Astola, J., Haavisto, P., Neuvo, Y.: Vector median filters. Proceedings of the IEEE 78(4), 678–689 (1990)

[4] Christensen, O.: Frames and Bases: An Introductory Course. Birkhäuser, Springer e-books (2008)

[5] Goutsias, J., Heijmans, H.J.A.M., Sivakumar, K.: Morphological Operators for Image Sequences. Comput. Vis. Image Und. 62(3), 326–346 (1995)

[6] van de Gronde, J.J., Roerdink, J.B.T.M.: Group-invariant colour morphology based on frames. IEEE Transactions on Image Processing (submitted)

[7] Mausfeld, R., Heyer, D. (eds.): Colour Perception: Mind and the physical world. Oxford University Press (November 2003)

[8] Serra, J.: Anamorphoses and function lattices. Image Algebra and Morphological Image Processing IV 2030(1), 2–11 (1993)

[9] 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)

[10] Talbot, H., Evans, C., Jones, R.: Complete Ordering and Multivariate Mathematical Morphology. In: Heijmans, H.J.A.M., Roerdink, J.B.T.M. (eds.) Mathematical Morphology and its Applications to Image and Signal Processing, pp. 27–34. Kluwer Academic Publishers (1998)

[11] Velasco-Forero, S., Angulo, J.: Mathematical Morphology for Vector Images Using Statistical Depth. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 355–366. Springer, Heidelberg (2011)

[6] Implementation available at `http://bit.ly/YarcHY`

Mathematical Morphology for Real-Valued Images on Riemannian Manifolds

Jesús Angulo[1] and Santiago Velasco-Forero[2]

[1] CMM-Centre de Morphologie Mathématique,
Mathématiques et Systèmes, MINES ParisTech, France
[2] ITWM - Fraunhofer Institute, Kaiserlautern, Germany
jesus.angulo@mines-paristech.fr, velascoforero@itwm.fraunhofer.de

Abstract. This paper introduces mathematical morphology for real-valued images whose support space is a Riemannian manifold. The starting point consists in replacing the Euclidean distance in the canonic quadratic structuring function by the Riemannian distance. Besides the definition of Riemannian dilation/erosion and Riemannian opening/closing, their properties are explored. We generalize also some theoretical results on Lasry–Lions regularization for Cartan–Hadamard manifolds. Theoretical connections with previous works on adaptive morphology and on manifold shape are considered. Various useful image manifolds are formalized, with an example using real-valued 3D surfaces.

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 $x \in E$ into a grey-level value in $\mathcal{T}$. 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 $\delta_b(f)(x) = \sup_{y \in E} \left(f(y) + b(x - y)\right)$, and $\varepsilon_b(f)(x) = \inf_{y \in E} \left(f(y) - b(y - x)\right)$; 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 [24,11,25]. Usually, the structuring function is a parametric family $b_\lambda(x)$, where $\lambda > 0$ is the scale parameter [12]. It is well known in the state-of-the-art of Euclidean morphology that the canonic family of structuring functions is the quadratic (or parabolic) one [19,7]; i.e., $b_\lambda(x) = -\frac{\|x\|^2}{2\lambda}$.

Aim of the Paper. Let us consider now that the support space is not Euclidean, this is case for instance if we deal with a smooth 3D surface, or more generally if the support space is a Riemannian manifold. The present work was inspired by the idea of Riemannian inf-convolution introduced in [4], which replaced the Euclidean distance in the canonic quadratic structuring function by the Riemannian distance. We adopt exactly the same starting point for the formulation of dilation/erosion in Riemannian manifolds (Section 2). Besides the definition of

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 279–291, 2013.

Riemannian dilation/erosion and Riemannian opening/closing, we explore their properties. We generalize also some theoretical results on Lasry–Lions regularization, with an original result for Cartan–Hadamard manifolds (more general than Moreau–Yosida regularization studied in [4]). We consider theoretically various useful case studies of image manifolds in Section 5, but due to the limited paper length, we only illustrate the case of real-valued 3D surfaces.

Related Work. Generalizations of Euclidean translation-invariant morphology have followed three main directions. On the one hand, adaptive morphology [18,26,1,8], where the structuring function becomes dependent on the position or the input image itself. Section 4 explores the connections of our framework with such approaches. On the second hand, the group morphology [22], where the translation invariance is replaces by other group invariance (similarity, affine, spherical, projective, etc.). Related to that, we have also the morphology for binary manifolds [21], whose relationship with our formulation is deeply studied in Section 3. Finally, we should cite also the classical notion of geodesic dilation [16] as the basic operator for (connective) geodesic reconstruction [25], where the marker image is dilated according to the metric yielded by the reference image (see also Section 4).

2 Riemannian Dilation and Erosion

Let us start by a formal definition of the two basic morphological operators for images supported on a Riemannian manifold.

Definition 1. *Let $\mathcal{M}$ a complete Riemannian manifold and $d_{\mathcal{M}} : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+$, $(x,y) \mapsto d_{\mathcal{M}}(x,y)$, is the geodesic distance on $\mathcal{M}$, for any image $f : \mathcal{M} \rightarrow \overline{\mathbb{R}}$, $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$, so $f \in \mathcal{F}(\mathcal{M}, \mathbb{R})$ and for $\lambda > 0$ we define for every $x \in \mathcal{M}$ the Riemannian dilation of f of scale parameter λ as*

$$\delta_\lambda(f)(x) = \sup_{y \in \mathcal{M}} \left\{ f(y) - \frac{1}{2\lambda} d_{\mathcal{M}}(x,y)^2 \right\} \tag{1}$$

and the Riemannian erosion of f of parameter λ as

$$\varepsilon_\lambda(f)(x) = \inf_{y \in \mathcal{M}} \left\{ f(y) + \frac{1}{2\lambda} d_{\mathcal{M}}(x,y)^2 \right\} \tag{2}$$

An obvious property of the Riemannian dilation and erosion is the *duality by the involution $f(x) \mapsto \complement f(x) = -f(x)$*, i.e., $\delta_\lambda(f) = \complement \varepsilon_\lambda(\complement f)$. As in classical Euclidean morphology, the adjunction relationship is fundamental for the construction of the rest of morphological operators.

Proposition 1. *For any two real-valued images defined on the same Riemannian manifold $\mathcal{M}$, i.e., $f, g : \mathcal{M} \rightarrow \overline{\mathbb{R}}$, the pair $(\varepsilon_\lambda, \delta_\lambda)$ is called a Riemannian adjunction*

$$\delta_\lambda(f)(x) \le g(x) \Leftrightarrow f(x) \le \varepsilon_\lambda(g)(x) \tag{3}$$

Proof. We have that the inequality $\delta_\lambda(f)(x) \leq g(x)$ means that

$$\sup_{y\in\mathcal{M}} \left\{ f(y) - \frac{1}{2\lambda} d_{\mathcal{M}}(x,y)^2 \right\} \leq g(x), \quad \forall x \in \mathcal{M},$$

It involves that $f(y) - \frac{1}{2\lambda} d_{\mathcal{M}}(x,y)^2 \leq g(x)$ for every $x, y \in \mathcal{M}$. This is equivalent to rewrite $f(y) \leq g(x) + \frac{1}{2\lambda} d_{\mathcal{M}}(x,y)^2$. Therefore, after substitution of $z = x$, we finally have

$$f(y) \leq \inf_{z\in\mathcal{M}} \left\{ g(y) + \frac{1}{2\lambda} d_{\mathcal{M}}(z,y)^2 \right\} = \varepsilon_\lambda(g)(y).$$

Hence, we have an adjunction if both images f and g are defined on the same Riemannian manifold $\mathcal{M}$, or in other terms, when the same "geodesic structuring function" $b_{x,\lambda}(y) = -\frac{1}{2\lambda} d_{\mathcal{M}}(x,y)^2$ is considered for each pixel. This result implies in particular that the Riemannian dilation commutes with the supremum and the Riemannian erosion with the infimum, i.e., for a given collection of images $f_i \in \mathcal{F}(\mathcal{M}, \overline{\mathbb{R}})$, $i \in I$, we have

$$\delta_\lambda \left(\bigvee_{i\in I} f_i \right) = \bigvee_{i\in I} \delta_\lambda(f_i); \quad \varepsilon_\lambda \left(\bigwedge_{i\in I} f_i \right) = \bigwedge_{i\in I} \varepsilon_\lambda(f_i).$$

In addition, using the classical results on adjunctions in complete lattices [11], we state that the products of the pair $(\varepsilon_\lambda, \delta_\lambda)$ lead to the adjoint opening and adjoint closing if and only the field of geodesic structuring functions is computed on a common manifold $\mathcal{M}$.

Definition 2. *Given an image $f \in \mathcal{F}(\mathcal{M}, \overline{\mathbb{R}})$, the Riemannian opening and Riemannian closing of scale parameter λ are respectively given by*

$$\gamma_\lambda(f)(x) = \sup_{z\in\mathcal{M}} \inf_{y\in\mathcal{M}} \left\{ f(y) + \frac{1}{2\lambda} d_{\mathcal{M}}(z,y)^2 - \frac{1}{2\lambda} d_{\mathcal{M}}(z,x)^2 \right\}, \tag{4}$$

and

$$\varphi_\lambda(f)(x) = \inf_{z\in\mathcal{M}} \sup_{y\in\mathcal{M}} \left\{ f(y) - \frac{1}{2\lambda} d_{\mathcal{M}}(z,y)^2 + \frac{1}{2\lambda} d_{\mathcal{M}}(z,x)^2 \right\}. \tag{5}$$

This technical point is very important since in some image manifold embedding the Riemannian manifold support $\mathcal{M}$ of image f depends itself on f. If $\mathcal{M}$ does not depends on f, the Riemannian opening and closing are respectively given by $\gamma_\lambda(f) = \delta_\lambda(\varepsilon_\lambda(f))$, and $\varphi_\lambda(f) = \varepsilon_\lambda(\delta_\lambda(f))$. We notice that this issue was already considered by Roerdink in [23] for the case of adaptive neighbourhood morphology.

Having the Riemannian opening and closing, all the other morphological filters defined by product of them are naturally defined, but keeping in mind that the geodesic structuring function should be fixed.

2.1 Properties of $\delta_\lambda(f)$ and $\varepsilon_\lambda(f)$

Other classical properties of Euclidean dilation and erosion have also the equivalent for Riemannian manifold $\mathcal{M}$, and they do not dependent on the geometry of $\mathcal{M}$.

Proposition 2. *Let $\mathcal{M}$ be a Riemannian manifold, and let $f, g \in \mathcal{F}(\mathcal{M}, \overline{\mathbb{R}})$ two real valued images $\mathcal{M}$. We have the following properties.*

1. *(Increaseness) If $f(x) \leq g(x)$, $\forall x \in \mathcal{M}$ then $\delta_\lambda(f)(x) \leq \delta_\lambda(g)(x)$ and $\varepsilon_\lambda(f)(x) \leq \varepsilon_\lambda(g)(x)$, $\forall x \in \mathcal{M}$ and $\forall \lambda > 0$.*
2. *(Extensivity and anti-extensivity) $\delta_\lambda(f)(x) \geq f(x)$ and $\varepsilon_\lambda(f)(x) \leq f(x)$, $\forall x \in \mathcal{M}$ and $\forall \lambda > 0$.*
3. *(Ordering property) If $0 < \lambda_1 < \lambda_2$ then $\delta_{\lambda_2}(f)(x) \geq \delta_{\lambda_1}(f)(x)$ and $\varepsilon_{\lambda_2}(f)(x) \leq \varepsilon_{\lambda_1}(f)(x)$.*
4. *(Invariance under isometry) If $T : \mathcal{M} \to \mathcal{M}$ is an isometry of $\mathcal{M}$ and if f is invariant under T, i.e., $f(Tz) = f(z)$ for all $z \in \mathcal{M}$, then the Riemannian dilation and erosion are also invariant under T, i.e., $\delta_\lambda(f)(Tz) = \delta_\lambda(f)(z)$ and $\varepsilon_\lambda(f)(Tz) = \varepsilon_\lambda(f)(z)$, $\forall z \in \mathcal{M}$ and $\forall \lambda > 0$.*
5. *(Extrema preservation) We have $\sup \delta_\lambda(f) = \sup f$ and $\inf \varepsilon_\lambda(f) = \inf f$, moreover if f is lower (resp. upper) semicontinuous then every minimizer (resp. maximizer) of $\varepsilon_\lambda(f)$ (resp. $\delta_\lambda(f)$) is a minimizer (resp. maximizer) of f, and conversely.*

Proof. By the properties of the Riemannian distance, 1., 2. and 3. are obvious. Property 4. is a generalization of the translation invariance, we have for the Riemannian dilation that

$$\begin{aligned}\delta_\lambda(f)(Tx) &= \sup_{y\in\mathcal{M}} \left\{ f(y) - \frac{1}{2\lambda} d_{\mathcal{M}}(Tx, y)^2 \right\} = \sup_{y\in\mathcal{M}} \left\{ f(Ty) - \frac{1}{2\lambda} d_{\mathcal{M}}(Tx, Ty)^2 \right\} \\ &= \sup_{y\in\mathcal{M}} \left\{ f(Ty) - \frac{1}{2\lambda} d_{\mathcal{M}}(x, y)^2 \right\} = \sup_{y\in\mathcal{M}} \left\{ f(y) - \frac{1}{2\lambda} d_{\mathcal{M}}(x, y)^2 \right\} = \delta_\lambda(f)(x);\end{aligned}$$

and similarly for the Riemannian erosion.

For 5., first, we have for the erosion that

$$\begin{aligned}\inf_{x\in\mathcal{M}} \varepsilon_\lambda(f)(x) &= \inf_{x\in\mathcal{M}} \inf_{y\in\mathcal{M}} \left\{ f(y) + \frac{1}{2\lambda} d_{\mathcal{M}}(x, y)^2 \right\} \\ &= \inf_{y\in\mathcal{M}} \inf_{x\in\mathcal{M}} \left\{ f(y) + \frac{1}{2\lambda} d_{\mathcal{M}}(x, y)^2 \right\} = \inf_{y\in\mathcal{M}} f(y).\end{aligned}$$

The rest of the proof of 5. is inspired from [4] as follows. Since $\varepsilon_\lambda(f) \leq f$ it is then obvious that every minimizer of f is a minimizer of $\varepsilon_\lambda(f)$ as well. Conversely, assume that f is lower semicontinuous, let x_0 be a minimizer of $\varepsilon_\lambda(f)$, i.e., $\varepsilon_\lambda(f)(x_0) = \inf_{z\in\mathcal{M}} \varepsilon_\lambda(f)(z) = \inf_{z\in\mathcal{M}}(f)(z)$. We need to prove that $f(x_0) = \inf_{z\in\mathcal{M}}(f)(z)$. Choose $x_{n n\in\mathbb{N}} \subset \mathcal{M}$ so that $f(x_n) + \frac{1}{2\lambda} d(x_0, x_n)^2 \leq \varepsilon_\lambda(f)(x_0) + \frac{1}{n}$; the we have

$$0 \leq \frac{1}{2\lambda} d(x_0, x_n)^2 \leq \varepsilon_\lambda(f)(x_0) + \frac{1}{n} - f(x_n) \leq \varepsilon_\lambda(f)(x_0) + \frac{1}{n} - \inf_{z \in \mathcal{M}} (f)(z) = \frac{1}{n} \to 0,$$

hence x_n converges to x_0, and since f is lower semicontinuous we get

$$f(x_0) \leq \lim_{n \to 0} f(x_n) \leq \lim_{n \to 0} \left(\varepsilon_\lambda(f)(x_0) + \frac{1}{n} \right) = \varepsilon_\lambda(f)(x_0) = \inf_{z \in \mathcal{M}} \varepsilon_\lambda(f)(z) = \inf_{z \in \mathcal{M}} f(z)$$

and x_0 is a minimizer of f. The proof for the dilation is similar.

Finally, as proved also in [4] we have the following property of localization which is useful for the practical computation on Riemannian dilation and erosion in local neighborhoods.

Proposition 3. *Let $\mathcal{M}$ be a Riemannian manifold, and let $f \in \mathcal{F}(\mathcal{M}, \overline{\mathbb{R}})$ an image satisfying that $f(x) \geq -\frac{c}{2}(1 + d(x, x_0)^2)$ (quadratically minorized) for some $c > 0$, $x_0 \in \mathcal{M}$. Then, for all $\lambda \in (0, \frac{1}{2c})$ and for all $\rho > \overline{\rho}$, we have that*

$$\varepsilon_\lambda(f)(x) = \inf_{y \in B(x,\rho)} \left\{ f(y) + \frac{1}{2\lambda} d_{\mathcal{M}}(x, y)^2 \right\}, \tag{6}$$

where

$$\overline{\rho} = \overline{\rho}(x, \lambda, c) = \sqrt{\lambda \frac{2f(x) + c(2d(x, x_0)^2 + 1)}{1 - 2\lambda c}},$$

and where the geodesic ball of center x and radius ρ is defined by $B(x, \rho) = \{y : d_{\mathcal{M}}(x, y) \leq \rho\}$. A similar result is obtained for the geodesic dilation $\delta_\lambda(f)(x)$, for $f(x) \leq \frac{c}{2}(1 + d(x, x_0)^2)$ (quadratically majorized), $c > 0$, where in this case $\overline{\rho} = \sqrt{\lambda \frac{c(2d(x,x_0)^2+1) - 2f(x)}{1 - 2\lambda c}}$.

This local character of geodesic operators, according to a quadratic growth assumption (weakly convexity/concavity), is similar to the case of Euclidean dilation/erosion using quadratic structuring functions [2].

For the Riemannian opening and closing, we have also the classical properties which are naturally proved as a consequence of the adjunction [11], namely: (i) *increaseness* of both operators; (ii) ordering relationships, i.e., $\gamma_{\lambda_2}(f)(x) \leq \gamma_{\lambda_1}(f)(x) \leq f(x) \leq \varphi_{\lambda_1}(f)(x) \leq \varphi_{\lambda_2}(f)(x)$ for $0 < \lambda_1 < \lambda_2$; (iii) *idempotency of both operators*, $\gamma_\lambda(\gamma_\lambda(f)) = \gamma_\lambda(f)$ and $\varphi_\lambda(\varphi_\lambda(f)) = \varphi_\lambda(f)$. The latter is fundamental for the theory of morphological scale-space filtering and the construction of granulometries [24,11] with respect to λ; the development of this point is out of the scope of this paper.

2.2 Weakly Convex Functions in a Cartan–Hadamard Manifold

As a generalization of the use of Moreau–Yosida regularization [20] (i.e., erosion by a quadratic structuring function), the Lasry–Lions regularization [17] is a theory of nonsmooth approximation for functions in Hilbert spaces using combinations of Euclidean dilation and erosion with quadratic structuring functions, which leads to the approximation of lower or upper semicontinuous functions by lipschitz continuous functions. As discussed in [4], the results of Moreau–Yosida regularization are extended to functions on a Cartan–Hadamard manifold [15]: simply connected complete Riemannian manifold of nonpositive sectional curvature, since its corresponding geodesic distance d is convex on all $\mathcal{M} \times \mathcal{M}$. That involves that [15] for every bounded subset B of $\mathcal{M}$, there exists $r > 0$ such that $x \mapsto d(x, B) = \inf d(x, y) : y \in B$ is convex in $B(x, r)$ and the set $B(x, r)$ is convex in $\mathcal{M}$, $\forall x \in B$; in addition $x \mapsto d(x, B)^2$ is C^1 smooth and convex.

Let us give a summary of our main result, which generalizes the case of convex functions in Cartan–Hadamard manifolds studied by [4] using the results for Hilbert spaces by [2] for the Lasry–Lions regularization.

Theorem 1. *Let $\mathcal{M}$ be a Riemannian manifold simply connected and with sectional curvature $K \leq 0$. Given two images $f, g \in \mathcal{F}(\mathcal{M}, \overline{\mathbb{R}})$, for all $0 < \mu < \lambda$ let us define the Lasry–Lions regularizers:*

$$(f_\lambda)^\mu(x) = \sup_{z \in \mathcal{M}} \inf_{y \in \mathcal{M}} \left\{ f(y) + \frac{1}{2\lambda} d_{\mathcal{M}}(z, y)^2 - \frac{1}{2\mu} d_{\mathcal{M}}(z, x)^2 \right\} \quad \textit{(pseudo-opening)}$$

$$(g^\lambda)_\mu(x) = \inf_{z \in \mathcal{M}} \sup_{y \in \mathcal{M}} \left\{ g(y) - \frac{1}{2\lambda} d_{\mathcal{M}}(z, y)^2 + \frac{1}{2\mu} d_{\mathcal{M}}(z, x)^2 \right\} \quad \textit{(pseudo-closing)}$$

We have $(f_\lambda)^\mu \leq f$ and $(g^\lambda)_\mu \geq g$. Assume that there exists $c, d > 0$, such that we have the following growing conditions:

$$f(x) \geq -\frac{c}{2}(1 + d(x, x_0)^2), \quad g(x) \leq \frac{d}{2}(1 + d(x, x_0)^2), \quad x_0 \in \mathcal{M}.$$

Then, for all $0 < \mu < \lambda < \frac{1}{c}$ the pseudo-opened image $(f_\lambda)^\mu$ is a C^1 function, for all $0 < \mu < \lambda < \frac{1}{d}$ the pseudo-closed image $(g^\lambda)_\mu$ is a C^1 function, whose gradient is Lipschitz continuous with constant $\max(1/\mu, 1/(\lambda - \mu))$.

In addition, if f is a lower (resp. g is a upper) semicontinuous function, the pseudo-opening of f (resp. pseudo-closing of g) converges uniformly to f (resp. g); i.e., $\lim_{\lambda,\mu \to 0} (f_\lambda)^\mu(x) = f(x)$ and $\lim_{\lambda,\mu \to 0} (g^\lambda)_\mu(x) = g(x)$.

The proof is skipped by limited length of paper. We notice that this is valid only for manifolds of nonpositive sectional curvature; the canonic example of negative curvature manifold is the hyperboloid; conversely, the canonic case of positively curved Riemannian manifold is the sphere, i.e., there are closed convex sets C of arbitrarily small diameter in $\mathbb{S}^2$ such that $x \mapsto d(x, C)$ is not convex on any neighborhood of C. Practical interest for quadratic "morphological lipschitzation" of surfaces or images valued on surfaces will be explored in ongoing work.

3 Connections with Roerdink's Manifold Morphology

In [21], Roerdink studied the formulation of dilation/erosion for binary images on smooth surfaces. But, before highlighting the connections with our framework, let us remind some basics on differential geometry for Riemannian manifolds [6].

The *tangent space* of the manifold $\mathcal{M}$ at a point $p \in \mathcal{M}$, denoted by $T_p\mathcal{M}$, is the set of all vectors tangent to $\mathcal{M}$ at p. The first issue to consider is how to transport vectors from one point of $\mathcal{M}$ to another. Let $p, q \in \mathcal{M}$ and let $\gamma : [a, b] \to \mathcal{M}$ be a parameterized curve (or path) from $\gamma(a) = p$ to $\gamma(b) = q$. For $\mathbf{v} \in T_p\mathcal{M}$, let $\mathbf{V}$ be the unique parallel vector field along γ with $\mathbf{V}(a) = \mathbf{v}$. The map $P_\gamma : T_p\mathcal{M} \to T_q\mathcal{M}$ determined by $P_\gamma(\mathbf{v}) = \mathbf{V}(b)$ is called *parallel transport from p to q along γ*, and $P_\gamma(\mathbf{v})$ the parallel translate of v along γ to q. Note that parallel transport from p to q is path dependent: the difference between two paths is a rotation around the normal to $\mathcal{M}$ at q. The *Riemannian distance* between two points p, $q \in \mathcal{M}$, denoted $d(p, q)$, is defined as the minimum length over all possible smooth curves between p and q. A *geodesic* $\gamma : [0, 1] \to \mathcal{M}$ connecting two points p, $q \in \mathcal{M}$ is the shortest path on $\mathcal{M}$ having elements p and q as endpoints. The geodesic curve $\gamma(t)$ can be specified in terms of a starting point $p \in \mathcal{M}$ and a tangent vector (initial constant velocity) $\mathbf{v} \in T_p\mathcal{M}$ as it represents the solution of Christoffel differential equation with boundary conditions $\gamma(0) = p$ and $\dot{\gamma}(0) = \mathbf{v}$. The idea behind *exponential map* Exp_p is to parameterize a Riemannian manifold $\mathcal{M}$, locally near any $p \in \mathcal{M}$, in terms of a mapping from the tangent space $T_p\mathcal{M}$ into a point in $\mathcal{M}$. The exponential map is injective on a zero-centered ball B in $T_p\mathcal{M}$ of some non-zero (possibly infinity) radius. Thus for a point q in the image of B under Exp_p there exists a unique vector $\mathbf{v} \in T_p\mathcal{M}$ corresponding to a minimal length path under the exponential map from p to q. Exponential maps may be associated to a manifold by the help of geodesic curves. The exponential map $\mathrm{Exp}_p : T_p\mathcal{M} \to \mathcal{M}$ associated to any geodesic $\gamma_\mathbf{v}$ emanating from p with tangent at the origin $\mathbf{v} \in T_p\mathcal{M}$ is defined as $\mathrm{Exp}_p(\mathbf{v}) = \gamma_\mathbf{v}(1)$, where the geodesic is given by $\gamma_\mathbf{v}(t) = \mathrm{Exp}_p(t\mathbf{v})$. The geodesic has constant speed equal to $\|d\gamma_\mathbf{v}/dt\|(t) = \|\mathbf{v}\|$, and thus the *exponential map preserves distances* for the initial point: $d(p, \mathrm{Exp}_p(\mathbf{v})) = \|\mathbf{v}\|$. A Riemannian manifold is geodesically complete if and only if the exponential map $\mathrm{Exp}_p(\mathbf{v})$ is defined $\forall p \in \mathcal{M}$ and $\forall \mathbf{v} \in T_p\mathcal{M}$. The inverse operator, named *logarithm map*, $\mathrm{Exp}_p^{-1} = \mathrm{Log}_p$ maps a point of $q \in \mathcal{M}$ into to their associated tangent vectors $\mathbf{v} \in T_p\mathcal{M}$. The exponential map is in general only invertible for a sufficient small neighbourhood of the origin in $T_p\mathcal{M}$, although on some manifolds the inverse exists for arbitrary neighbourhoods. For a point q in the domain of Log_p the *geodesic distance* between p and q is given by $d(p, q) = \| \mathrm{Log}_p(q)\|$.

The idea behind the binary Riemannian morphology on smooth surfaces [21] is to replace the translation invariance by the parallel transport (the transformations are referred to as "covariant" operations). Let $\mathcal{M}$ be a (geodesically complete) Riemannian manifold and $\mathcal{P}(\mathcal{M})$ denotes the set of all subsets of M. A binary image X on the manifold is just $X \in \mathcal{P}(\mathcal{M})$. Let $A \subset \mathcal{M}$ be the basic structuring, a subset which is defined on the tangent space at a given point ω of $\mathcal{M}$ by $\tilde{A} = \mathrm{Log}_\omega(A) \subset T_\omega\mathcal{M}$. Let $\gamma = \gamma_{[p,q]}$ be a path from p to q, then the operator

$$\tau_\gamma(A) = \mathrm{Exp_q}\, P_\gamma\, \mathrm{Log_p}(A) = B,$$

transports the subset A of p to the set B of q. As the image of the set X under parallel translation from p to q will depend in general on which path is taken;

the solution proposed in [21], denoted by δ_A^{Roerd}, is to consider all possible paths from p to q. The mapping $\delta_A^{Roerd} : \mathcal{P}(\mathcal{M}) \to \mathcal{P}(\mathcal{M})$ given by

$$\delta_A^{Roerd}(X) = \bigcup_{x\in\mathcal{M}} \bigcup_{\gamma} \tau_\gamma(A) = \bigcup_{x\in\mathcal{M}} \bigcup_{\gamma} \mathrm{Exp}_{\mathrm{x}}\, P_{\gamma_{[\omega,x]}} \operatorname{Log}_\omega(A), \tag{7}$$

is a dilation of image X according to the structuring element A. Using the symmetry group morphology [22], this operator can be rewritten as $\delta_A^{Roerd}(X) = \bigcup_{x\in\mathcal{M}} \mathrm{Exp}_x\, P_{\gamma_{[\omega,x]}} \operatorname{Log}_\omega(\bar{A})$, where $\bar{A} = \bigcup_{s\in\Sigma} sA$, with Σ being the holonomy group around the normal at ω. For instance, if $\tilde{A} = \operatorname{Log}_\omega(A)$ is a line segment of length r starting at ω then $\bar{A}$ is a disk of radius r centered at ω.

Coming back to our framework, we notice that working on a geodesically complete Riemannian manifold $\mathcal{M}$, our dilation and erosion for an image $f \in \mathcal{F}(\mathcal{M},\overline{\mathbb{R}})$ can be rewritten as

$$\delta_\lambda(f)(x) = \sup_{y\in\mathcal{M}} \left\{ f(y) - \frac{1}{2\lambda}\|\operatorname{Log}_{\mathrm{x}}(y)\|^2 \right\},\ \ \varepsilon_\lambda(f)(x) = \inf_{y\in\mathcal{M}} \left\{ f(y) + \frac{1}{2\lambda}\|\operatorname{Log}_{\mathrm{x}}(y)\|^2 \right\}.$$

If we replace the geodesic structuring element function $b_{x,\lambda}(y) = -\frac{\|\operatorname{Log}_x(y)\|}{2\lambda}$ by a flat geodesic structuring element by defining

$$b_{x,\lambda}(y) = \begin{cases} 0 & y \in B_\lambda(x) \\ -\infty & y \notin B_\lambda(x) \end{cases}$$

where $B_\lambda(x) = \{y : \|\operatorname{Log}_x(y)\| \leq \lambda\}$ is the geodesic ball centered at x of radius λ, we obviously have that the corresponding *Riemannian flat dilation and erosion of size* λ can be rewritten as:

$$\delta_{B_\lambda}(f)(x) = \sup\left\{f(y) :\ \ y \in B_\lambda(x)\right\}, \varepsilon_{B_\lambda}(f)(x) = \inf\left\{f(y) :\ \ y \in B_\lambda(x)\right\}. \tag{8}$$

From our viewpoint, if we consider the geodesic structuring element at a given pixel $\omega \in \mathcal{M}$; i.e.,$B_\lambda(\omega)$, as the reference structuring element of Roerdink, i.e., $A = B_\lambda(\omega)$, we can now formulate the *ω-driven Riemannian flat dilation and erosion* at scale λ as

$$\breve{\delta}_\lambda^\omega(f)(x) = \sup\left\{ f(y) :\ \ y \in \mathrm{Exp}_{\mathrm{x}}\, P_{\gamma^{geo}_{[\omega,x]}} \operatorname{Log}_\omega\left(B_\lambda(\omega)\right)\right\}, \tag{9}$$

$$\breve{\varepsilon}_\lambda^\omega(f)(x) = \inf\left\{ f(y) :\ \ y \in \mathrm{Exp}_{\mathrm{x}}\, P_{\gamma^{geo}_{[\omega,x]}} \operatorname{Log}_\omega\left(B_\lambda(\omega)\right)\right\}. \tag{10}$$

However, with respect to dilation (7), we prefer to consider in our case that the parallel transport from ω to x is done exclusively along the geodesic path $\gamma^{geo}_{[\omega,x]}$ between ω and x, i.e., if $\operatorname{Log}_\omega\left(B_\lambda(\omega)\right)$ was a line in ω then it will be also at x a line, but rotated.

4 Connections with Classical Euclidean Morphology

Riemannian dilation and erosion generalize previous operators in Euclidean morphology for both, translation invariant and spatially-variant morphology.

Spatially-invariant Operators. First of all, it is obvious that the Riemannian dilation/erosion naturally extends the quadratic Euclidean dilation/erosion for images $\mathcal{F}(\mathbb{R}^d, \overline{\mathbb{R}})$ by considering that the intrinsic distance is the Euclidean one (or the discrete one for $\mathbb{Z}^d$), i.e., $d_{\mathcal{M}}(x, y) = \|x - y\| = d_{space}(x, y)$. By the way, we note also that the definition of the Riemannian flat dilation and erosion of size λ given in (8) is compatible with the formulation of the classical geodesic dilation and erosion [16] of size λ of image f (marker) constrained by the image g (reference or mask), $\delta_{g,\lambda}(f)$ and $\varepsilon_{g,\lambda}(f)$, which underly the operators by reconstruction [25], where the upper-level sets of the reference image g are considered as the manifold $\mathcal{M}$ where the geodesic distance is defined.

Spatially-variant (adaptive) Operators. From [13], the idea of embedding a grey-level image $f \in \mathcal{F}(\mathbb{R}^2, \overline{\mathbb{R}})$, $x = (x_1, x_2)$, into a surface embedded in $\mathbb{R}^3$, i.e., $f(x) \mapsto \xi_x = (x_1, x_2, \alpha f(x_1, x_2))$, $\alpha > 0$ (scaling parameter useful for controlling intensity distances). This embedded Riemannian manifold $\mathcal{M} = \mathbb{R}^2 \times \overline{\mathbb{R}}$ has a product metric of type $ds^2_{\mathcal{M}} = ds^2_{space} + \alpha ds^2_f$, where $ds^2_{space} = dx_1^2 + dx_2^2$ and $ds^2_f = df^2$. The geodesic distance between two points ξ_x, $\xi_y \in \mathcal{M}$ is the length of the shortest path between the points, i.e., $d_{\mathcal{M}}(\xi_x, \xi_y) = \min_{\gamma=\gamma_{[\xi_x,\xi_y]}} \int_\gamma ds_{\mathcal{M}}$. As shown in [26], this is the framework behind the morphological amoebas [18], which are flat spatially adaptive structuring functions centered in a point x, $A_\lambda(x)$, computed by thresholding the geodesic distance at radius $\lambda > 0$, i.e., $A_\lambda(x) = \{y \in E : \, d_{\mathcal{M}}(\xi_x, \xi_y) < \lambda\}$. In the discrete setting, it is given by

$$d_{\mathcal{M}}(\xi_x, \xi_y) = \min_{\{\xi^1=\xi_x, \xi^2, \cdots, \xi^N=\xi_y\}} \sum_{i=1}^{N} \sqrt{(x_1^i - x_1^{i+1})^2 + (x_2^i - x_2^{i+1})^2} + \alpha |f(x^i) - f(x^{i+1})|.$$

We should remark that for $x \to y$ and smooth manifold, the geodesic distance is asymptotically equivalent to the corresponding distance in the Euclidean product space, i.e., $d_{\mathcal{M}}(\xi_x, \xi_y)^2 \approx d_{space}(x, y)^2 + \alpha^2 |f(x) - f(y)|^2$, which is the distance appearing in the bilateral structuring functions [1]. We can also see that the salience maps behind the salience adaptive structuring elements [8] can be approached in a Riemannian formulation.

5 Various Useful Case Studies

We state in this Section several interesting cases of image manifolds (and their corresponding Riemannian distances) which can be processed using Riemannian morphological operators.

Embedding an Euclidean Image into the Structure Tensor Manifold. Besides the embedding discussed in previous Section, we can consider other more geometric embedding, using for instance the local structure. More precisely, given a 2D Euclidean image $f(x) = f(x_1, x_2) \in \mathcal{F}(\mathbb{R}^2, \overline{\mathbb{R}})$, the structure tensor representing the local orientation and edge information [10] is obtained by Gaussian smoothing of the dyadic product $\nabla f \nabla f^T$:

$$S(f)(x) = G_\sigma * \left(\nabla f(x_1,x_2)\nabla f(x_1,x_2)^T\right) = \begin{pmatrix} s_{x_1x_1}(x_1,x_2) & s_{x_1,x_2}(x_1,x_2) \\ s_{x_1x_2}(x_1,x_2) & s_{x_2x_2}(x_1,x_2) \end{pmatrix}$$

where $\nabla f(x_1,x_2) = \left(\frac{\partial f(x_1,x_2)}{\partial x_1}, \frac{\partial f(x_1,x_2)}{\partial x_2}\right)^T$ is the 2D spatial intensity gradient and G_σ stands for a Gaussian smoothing with a standard deviation of σ. From a mathematical viewpoint, $S(f)(x) : E \to \mathrm{SPD}(2)$ is an image where at each pixel we have a symmetric positive definite matrix 2×2. The differential geometry in the manifold $\mathrm{SPD}(n)$ is very well-known [5]. Namely, the metric is given by $ds^2_{SPD(n)} = \mathrm{tr}(M^{-1}dM)$ and the Riemannian distance is defined as $d_{SPD(n)}(M_1,M_2) = \|\log\left(M_1^{-1/2}M_2M_1^{-1/2}\right)\|_F$, $\forall M_1, M_2 \in \mathrm{SPD}(n)$. Let consider now the embedding $f(x) \mapsto \xi_x = (x_1, x_2, \alpha S(f)(x_1,x_2))$, $\alpha > 0$, in the product manifold $\mathcal{M} = \mathbb{R}^2 \times \mathrm{SPD}(2)$, which has the product metric $ds^2_{\mathcal{M}} = ds^2_{space} + \alpha ds^2_{SPD(2)}$. It is a (complete, not compact, negative sectional curved) Riemannian manifold of geodesic distance $d_{\mathcal{M}}(\xi_x,\xi_y) = \min_{\gamma_{\xi_x,\xi_y}} \int_\gamma ds_{\mathcal{M}}$, which is asymptotically equal to $d_{\mathcal{M}}(\xi_x,\xi_y)^2 \approx d_{space}(x,y)^2 + \alpha d_{SPD(2)}(S(f)(x), S(f)(y))^2$. By means of this embedding, we can compute anisotropic morphological operators following the flow coherence of image structures.

Embedding an RGB-D Image into the $(x, y, Depth, f_R, f_G, f_B)$ **Manifold.** Current technologies of range cameras, such as the popular Kinect one, produce RGB images $f_{RGB} \in \mathcal{F}(\mathbb{R}^2, \overline{\mathbb{R}}^3)$ together with a depth map (or distance image) $f_{Depth} \in \mathcal{F}(\mathbb{R}^2, \overline{\mathbb{R}})$. The three components of the colour image can be morphologically processed using the following embedding: $(f_{RGB}(x), f_{Depth}(x)) \mapsto \xi_x = (x_1, x_2, \alpha f_R(x), \alpha f_G(x), \alpha f_B(x), \beta f_{Depth}(x))$, $\alpha, \beta > 0$. The Riemannnian manifold $\mathcal{M}$ is embedded in $\mathbb{R}^6$ and has the metric $ds^2_{\mathcal{M}} = ds^2_{space} + \alpha ds^2_{RGB} + \beta ds^2_{Depth}$, with $ds^2_{RGB} = df_R^2 + df_G^2 + df_B^2$ and $ds^2_{Depth} = df^2_{Depth}$; the geodesic distance is asymptotically $d_{\mathcal{M}}(\xi_x,\xi_y)^2 \approx d_{space}(x,y)^2 + \alpha d_{RGB}(x,y)^2 + \beta d_{Depth}(x,y)^2$. Working in this manifold, the 3D information from the scene 2D × Depth is used to adaptively constraint the operators on the colour images.

Embedding Star-like 3D Surfaces into the Sphere $\mathbb{S}^2$**.** Let consider a smooth surface $\mathcal{S} \subset \mathbb{R}^3$ under the assumption that it corresponds to a star-shaped 3D objet: that means that there exists a point $z_0 \in \mathbb{R}^3$ within the object, such that each ray originating from this point intersects the object's surface exactly one; this point is denoted $\pi_{z_0 \to \mathcal{S}}$. We can assign to each ray, representing a 3D orientation in space and therefore an element of the sphere $\mathbb{S}^2$, the distance of the ray-surface-intersection to the origin of the ray. Hence, we have a function $f : \mathbb{S}^2 \to \mathbb{R}_+$, where $(\mathcal{S}, z_0) \mapsto f(\theta,\phi) = \|z_0 - \pi_{z_0\to\mathcal{S}}\|$, (θ,ϕ) are the standard spherical coordinates. The Riemannian geodesic distance in $\mathbb{S}^2$ (or Great circle distance) is simple the angle between the two points; i.e., $d_{\mathbb{S}^2}(\xi_1,\xi_2) = \arccos(\xi_1 \cdot \xi_2)$, where $\xi_i = (\sin\theta_i \cos\varphi_i, \sin\theta_i \sin\varphi_i, \cos\theta_i)$. The image f can be processed using the Riemannian morphological operators, obtaining the image f' and the corresponding processed surface $\mathcal{S}'$ is simply obtained by inversion of projective mapping $\pi_{z_0\to\mathcal{S}'}$.

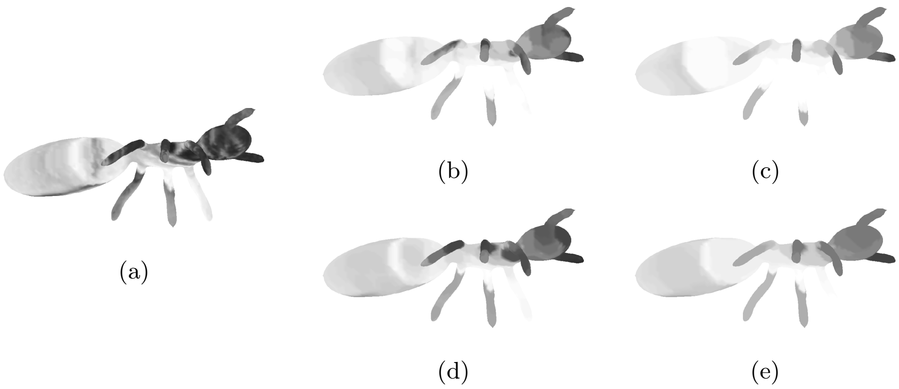


Fig. 1. Morphological processing of real valued 3D surface: (a) original image on a surface $\mathcal{S} \subset \mathbb{R}^3$, $f(x) \in \mathcal{F}(\mathcal{S}, \mathbb{R}_+)$; (b) and (c) Riemannian dilation $\delta_\lambda(f)(x)$ with respectively $\lambda = 4$ and $\lambda = 8$; (d) and (e) Riemannian closing $\varphi_\lambda(f)(x)$ with respectively $\lambda = 4$ and $\lambda = 8$

Morphological Processing of Real Valued 3D Surfaces. In Fig. 1(a) is given an example of real-valued 3D surface, i.e., the image to be processed is $f : \mathcal{S} \subset \mathbb{R}^3 \to \overline{\mathbb{R}}$. In practice, the 3D surface is represented by a mesh (i.e., triangulated manifold with a discrete structure composed of vertices, edges and faces). In our example, the grey-level intensities are supported on the vertices. In the case of a discrete approximation of a manifold based on mesh representation, the geodesic distance $d_\mathcal{S}(x, y)$ can be calculated by the Floyd–Warshall algorithm for finding shortest path in the weighted graph of vertices of the mesh. Efficient algorithms are based on Fast Marching generalized to arbitrary triangulations [14]. Fig. 1 depicts examples of Riemannian dilation $\delta_\lambda(f)$ and Riemannian closing $\varphi_\lambda(f)$, for two different scales ($\lambda = 4$ and $\lambda = 8$).

6 Perspectives

This work on Riemannian mathematical morphology can be continued in various ways. On the one hand, some points require a more deeply technical analysis: granulometric framework for Riemannian openings/closings and other scale-space properties, regularization properties of other morphological filters than the pseudo-openings/closings, formulation of levelings and other connected operators for Riemannian manifolds, etc. On the other hand, a deeper exploration of the flat Riemannian operators, associated to a fixed geodesic structuring element, inspired from the Roerdink idea of parallel transport, will be done considering for instance notions from optimal transport. Finally, Riemannian morphological framework can be also approached from a continuous viewpoint using the corresponding Hamilton-Jacobi equation. In particular, the starting point can be the study by Dragoni [9] (see also [3]), which considers the Hamilton-Jacobi equation $u_t + \frac{1}{2}|\sigma(x)Du|^2 = 0$, $(x,t) \in \mathbb{R}^d \times \mathbb{R}_+$, $u(x,0) = f(x)$ where $\sigma(x)$ is a Carnot–Carathéodory metric (naturally defined in Riemannian manifolds), whose viscosity solution is $u(x,t) = \inf_{y \in \mathbb{R}^d} \left\{ f(y) + \frac{d(x,y)^2}{2t} \right\}$, with $d(x,y)$ being the Carnot–Carathéodory distance. In practice, that involves the construction of stable intrinsic numerical methods for Hamilton-Jacobi equation on Riemannian manifolds, and in particular to 3D surfaces.

References

1. Angulo, J.: Morphological Bilateral Filtering and Spatially-Variant Adaptive Structuring Functions. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 212–223. Springer, Heidelberg (2011)
2. Attouch, D., Aze, D.: Approximation and regularization of arbitray functions in Hilbert spaces by the Lasry-Lions method. Annales de l'I.H.P., section C 10(3), 289–312 (1993)
3. Azagra, D., Ferrera, J., López-Mesas, F.: Nonsmooth analysis and Hamilton-Jacobi equations on Riemannian manifolds. J. Funct. Anal. 220(2), 304–361 (2005)
4. Azagra, D., Ferrera, J.: Inf-Convolution and Regularization of Convex Functions on Riemannian Manifols of Nonpositive Curvature. Rev. Mat. Complut. 19(2), 323–345 (2006)
5. Bhatia, R.: Positive Definite Matrices. Princeton University Press (2007)
6. Berger, M., Gostiaux, B.: Differential Geometry: Manifolds, Curves, and Surfaces. Springer (1987)
7. van den Boomgaard, R., Dorst, L.: The morphological equivalent of Gaussian scale-space. In: Proc. of Gaussian Scale-Space Theory, pp. 203–220. Kluwer (1997)
8. Ćurić, V., Luengo Hendriks, C.L., Borgefors, G.: Salience adaptive structuring elements. IEEE Journal of Selected Topics in Signal Processing 6(7), 809–819 (2012)
9. Dragoni, F.: Limiting behavior of solutions of subelliptic heat equations. Nonlinear Differential Equations and Applications 14(3), 429–441 (2007)
10. Förstner, W., Gülch, E.: A fast operator for detection and precise location of distinct points, corners and centres of circular features. In: Proc. of ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data, pp. 281–304 (1987)
11. Heijmans, H.J.A.M.: Morphological image operators. Academic Press, Boston (1994)
12. Jackway, P.T., Deriche, M.: Scale-Space Properties of the Multiscale Morphological Dilation-Erosion. IEEE Trans. Pattern Anal. Mach. Intell. 18(1), 38–51 (1996)
13. Kimmel, R., Sochen, N., Malladi, R.: Images as embedding maps and minimal surfaces: movies, color, and volumetric medical images. In: Proc. of IEEE CVPR 1997, pp. 350–355 (1997)
14. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. Proc. of National Academy of Sci. 95(15), 8431–8435 (1998)
15. Lang, S.: Fundamentals of differential geometry. Springer (1999)
16. Lantuejoul, C., Beucher, S.: On the use of the geodesic metric in image analysis. Journal of Microscopy 121(1), 39–49 (1981)
17. Lasry, J.M., Lions, P.-L.: A remark on regularization in Hilbert spaces. Israel Journal of Mathematics 55, 257–266 (1986)
18. Lerallut, R., Decencière, E., Meyer, F.: Image filtering using morphological amoebas. Image and Vision Computing 25(4), 395–404 (2007)

19. Maragos, P.: Slope Transforms: Theory and Application to Nonlinear Signal Processing. IEEE Trans. on Signal Processing 43(4), 864–877 (1995)
20. Moreau, J.J.: Fonctionnelles convexes. Lecture notes, Collége de France, Paris (1967), http://archive.numdam.org
21. Roerdink, J.B.T.M.: Manifold shape: from differential geometry to mathematical morphology. In: Shape in Picture, NATO ASI F, vol. 126, pp. 209–223. Springer (1994)
22. Roerdink, J.B.T.M.: Group morphology. Pattern Recognition 33, 877–895 (2000)
23. Roerdink, J.B.T.M.: Adaptivity and group invariance in mathematical morphology. In: Proc. of ICIP 2009 (2009)
24. Serra, J.: Image Analysis and Mathematical Morphology, vol. II: Theoretical Advances. Academic Press, London (1988)
25. Soille, P.: Morphological Image Analysis. Springer, Berlin (1999)
26. Welk, M., Breuß, M., Vogel, O.: Morphological amoebas are self-snakes. Journal of Mathematical Imaging and Vision 39(2), 87–99 (2011)

A Weight Sequence Distance Function

Benedek Nagy[1,⋆], Robin Strand[2], and Nicolas Normand[3]

[1] Faculty of Informatics, University of Debrecen, Hungary
[2] Centre for Image Analysis, Uppsala University, Sweden
[3] LUNAM Université, Université de Nantes, IRCCyN UMR CNRS 6597, Nantes, France
robin@cb.uu.se, nbenedek@inf.unideb.hu

Abstract. In this paper, a family of weighted neighborhood sequence distance functions defined on the square grid is presented. With this distance function, the allowed weight between any two adjacent pixels along a path is given by a weight sequence. We build on our previous results, where only two or three unique weights are considered, and present a framework that allows any number of weights. We show that the rotational dependency can be very low when as few as three or four unique weights are used. An algorithm for computing the distance transform (DT) that can be used for image processing applications is also presented.

1 Introduction

In a digital space (given for example by the pixels on a computer screen), not all properties of the Euclidean geometry are fulfilled. This is mainly due to the discrete (as opposed to continuous) structure of digital spaces. An example that shows that the Euclidean *distance function* has some disadvantages in digital spaces is the following: Circles, i.e., points of equal distance form a single point, are in general not connected in the usual sense: The eight points at distance $\sqrt{5}$ from a given point are disconnected with any of the common digital connectivities. Moreover, for most of the positive real values r there is not any point with this distance from any other point, see for example [6]. Also, computing with the Euclidean distance [11], in some cases, is very time consuming, especially for the so-called constrained distance transform, see the discussion in [14].

Therefore, the use of digital, i.e., path-based, distances is potentially very important for both digital image processing and computer graphics. For a path-based distance, each distance value attained is the cost of a connected path between two pixels in a square grid. In this aspect, the digital approach we follow is fundamentally different from the approach when Euclidean distances are computed. We believe that it is important to develop both the theory based on these digital distances and practical algorithms for image processing that effectively can utilize these distances.

The two digital distances first described in the literature are the city block and chessboard distances [12]. It is very easy to compute and use them, but they have very high rotational dependency (anisotropy). The theory of digital distances has developed rapidly

[⋆] The work of Benedek Nagy is supported by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 292–301, 2013.

from 1980's. The weighted (chamfer) distances [1], where the grid points together with costs to local neighbors form a graph in which the minimal cost path is the distance, is a well-known and often used concept in image processing. Contrary to weighted distances, the allowed steps may vary along the path with distances based on neighborhood sequences from a predefined set of steps [3]. For instance, by mixing the city block and chessboard neighborhood. In [18,16], the concept of weighted distances is generalized by allowing the size of the neighborhood to vary along the path. In this way, we get a distance function with potentially lower rotational dependency compared to when a fixed neighborhood is used. Recent results on Euclidean distance approximation are found in [4,2,7].

In this paper, we extend the idea presented in [15], where distances defined by three different local steps were considered. We allow using a fixed, but arbitrary large, number of possible local steps in the neighborhood sequence. Each of the allowed steps use only the 8-neighborhood of the pixels, but with different weights. Our main motivation is to provide a framework, to define digital distance functions that have as low rotational dependency as possible, that can be used to develop efficient image processing tools. Here, this is obtained by finding weight sequences that approximate the Euclidean distance.

Given a distance function, a *distance transform* is a transform where each element in a set is assigned the distance to the closest (as given by the distance function) element in a complementary set. The result of a distance transform is called a distance map. This tool is often used in image processing and computer graphics [5]. In digital geometry, the geometry of integer grids is used for building algorithms for, for example, image processing. It is very natural to define distance functions in this setting by minimal cost paths. In this paper, we present an algorithm for computing the distance map.

The structure of the paper is as follows. In the next section we present definitions and also some theoretical results, e.g., a formula to compute the point-to-point distance. In Section 3 parameter optimization is shown to obtain distances with low rotational dependency, i.e., approximating the Euclidean distance in this sense. Section 4 contains an algorithm for computing the distance map (DM) with some examples.

2 Theory

In this paper, we consider grid points with integer coordinates. Of course, in image processing, each grid point is associated with a picture element, pixel. In a city block (resp. chessboard distance), points with unit difference in at most one (resp. two) of the coordinates have unit distance. Here, we use the notion of 1- and 2-neighbors in the following sense: Two grid points $P_1 = (x_1, y_1), P_2 = (x_2, y_2) \in \mathbb{Z}^2$ are ρ-neighbors, $\rho \in \{1, 2\}$, if

$$\begin{aligned} |x_1 - x_2| + |y_1 - y_2| \le \rho \text{ and} \\ \max\{|x_1 - x_2|, |y_1 - y_2|\} = 1. \end{aligned} \tag{1}$$

The points are *strict* ρ-neighbors if the equality in (1) is attained. Two points are *adjacent* if they are 2-neighbors. A neighborhood sequence (ns) B is a sequence

$B = (b(i))_{i=1}^{\infty}$ of neighborhood relations [3,8]. The shortest B-path between any two points can be computed by a greedy algorithm (see, e.g., [8]). A formula to compute B-distances can be found in [9,10] for the square grid. However, in this paper we use weight sequences instead of neighborhood sequences and will therefore use a modified description. A weight sequences is denoted $W = (w(i))_{i=1}^{\infty}$. A weight sequence W can be used as follows: In a W-distance, opposed to the B-distance, a weight to the 2-neighbors are always given by the weight sequence. The cost of a move to a 1-neighbor is 1 in every step, and the cost of a move to a strict 2-neighbor is given in the weight sequence.

In this work, we allow $m \in \mathbb{N},\ m \geq 2$ different neighborhood relations:

- a traditional 1-step is a step between 1-neighbors with unit weights, the sign ∞ denote these steps in W (practically, strict 2-steps are not allowed);
- a traditional 2-step is a step between 2-neighbors with unit weights, they are denoted by 1 in W;

and if $m > 2$, then let $\{w_3, \ldots, w_m\}$ be the used weight set and in these cases the further steps are:

- weighted 2-steps are steps between 1-neighbors with unit weights, and between strict 2-neighbors are steps with a weight w_k (where $3 \leq k \leq m$ and $1 \leq w_k \leq \infty$).

In this paper the weight sequence W can contain m weights of a predefined weight set, i.e., $w(i) \in \{1, \infty, w_3, \ldots, w_m\}$ for all $i \in \mathbb{N}, i > 0$.

A *path* in a lattice is a sequence of adjacent lattice points. A path $P_0, P_1, \ldots, P_n$ is a path of n steps where for all $i \in \{1, 2, \ldots, n\}$, P_{i-1} and P_i are adjacent.

The cost (weighted length) of a path is the sum of the weights along the path, i.e.,

$$\sum_{i=1}^{n} \delta_i, \text{ where } \delta_i = \begin{cases} w(i), & \text{if } P_{i-1} \text{ and } P_i \text{ are strict} \\ & \text{2-neigbors;} \\ 1, & \text{otherwise.} \end{cases}$$

When the weight sequence W is fixed we use the term W-path for paths having finite cost as defined by the weight sequence W. A W-path between P and Q is a *minimal cost* W-path if no other W-path between the points has lower cost. (If a step with weight ∞ has been taken, then the length of this path is ∞ and it is greater than any finite number.) The W-distance between P and Q is the cost of a minimal cost W-path between P and Q.

Regarding only the W-distances and paths with minimal costs they are obtained without any step with a weight value $w_i > 2$. This fact allows reducing our notation, the steps and so the values in the weight sequence W with weight ∞ (together with all values that are larger than 2) can be replaced by the same number (and it could be any number that is larger than 2). We use the notation ∞ for these steps in this paper. Based on this we can say that in our paths only weights between 1 and 2 play important role.

Example 1. Let the weight sequence $W = (1, 1.9, 1.8, 1, 1.5, \ldots)$. Then the shortest W-path from $(0, 0)$ to $(2, 2)$ includes two diagonal steps with weights $1 + 1.9 = 2.9$.

However the shortest W-paths from $(0, 0)$ to $(3, 3)$ is not a continuation of the former path, but consists of a diagonal step to $(1, 1)$ with weight 1, then two consecutive 1-steps (to either $(1, 2)$ or $(2, 1)$ and, then, to $(2, 2)$) and finally a diagonal step with weight 1: in this way the W-distance of $(0, 0)$ and $(3, 3)$ is 4. Comparing these shortest paths with four steps, one can reach the point $(3, 3)$ in three steps from $(0, 0)$, but the weight of these three diagonal steps 4.7 together.

The W-distance of $(0, 0)$ and $(2, 3)$ is 3.8 and it comes from the shortest path including a diagonal step (weight 1), then, we need a 1-step and diagonal step by weight 1.8.

By Example 1, one can see that greedy algorithm cannot be used to provide shortest paths. If a smaller weight appears after a larger weight in W, we may need this smaller in our shortest path, but it depends on both the weight sequence and on the difference of the coordinate values of the points.

2.1 Formula for Computing the Distance Function

Now we give a formula for computing the distance between any two grid points. The formula is used for finding optimal parameters in Section 3.

Let the weight sequence W, with the the weight set $\{1, \infty, w_3, \ldots, w_m\}$ and the point $(x, y) \in \mathbb{Z}^2$, where $x \geq y \geq 0$, be given. The number of steps in an optimal path from the point $\mathbf{0}(0, 0)$ to the point (x, y) is between x and $x + y$ since the last case gives only 1-steps, which are always allowed. In this case, the distance is exactly $x + y$. The first case gives $x - y$ 1-steps and y 2-steps, so to get the path cost, we sum up the 1-steps and the y smallest weights of the first x elements. In the general case, we find the optimal value of $f = 0 \ldots y$ (that gives the number of 2-steps) by summing up the $x - y + 2f$ 1-steps and the $y - f$ smallest weight among the first $x + f$ elements in the weight sequence (2-steps). The distance is defined for the f that gives the path with the lowest cost. This gives the formula

$$d(\mathbf{0}, (x, y); W) = \min_{f=0..y} \left\{ x - y + 2f + \sum_{i \in I} w(i) \right\} \tag{2}$$

where the index set I contains the index of the smallest $y - f$ weight values among the first $x + f$ values of the weight sequence W. Since the roles of the x- and y-coordinate are similar, and our distance function is translation invariant, one can easily compute the W-distance of any pair of points of $\mathbb{Z}^2$ by our formula.

Our general approach consists several special cases:

- $W = (w_3)_1^\infty$ – traditional chamfer distance
- W contains only 1's and ∞'s – traditional distances based on neighborhood sequences ([3,8,9])
- W contains only 1's and w_3 – distances defined by weighted neighborhood sequences ([14])
- W contains only values 1, ∞, w_3 – distances defined by three types of local steps [15]

Note that, as opposed to the above mentioned first three usual cases, in our general case the greedy algorithm do not produce the optimal path and so the distance cannot be obtained by their help.

3 Parameter Optimization

In this section, we give some results on the approximation of the Euclidean distance. We find weight sequences that give a small difference between the Euclidean distance $d_E(\cdot,\cdot)$ and the weight sequence distance $d(\cdot,\cdot;W)$ between the point **0** and the point (x, y), where $x \geq y \geq 0$. See also [4]. The general case follows by symmetry as we discussed at the previous section about the formula.

Lemma 1. *If the weight sequence W is non-decreasing and all elements in W are smaller than or equal to 2, the distance in (2) is given by*

$$d(\mathbf{0},(x,y);W) = x - y + \sum_{i=1}^{y} w(i)$$

Proof. Since the lowest weights are the first in the weight sequence, we have

$$\min_{f=0..y}\left\{x - y + 2f + \sum_{i\in I} w(i)\right\} = \min_{f=0..y}\left\{x - y + 2f + \sum_{i=1}^{y-f} w(i)\right\},$$

where the sum from $i = 1$ to $i = 0$ is 0. Since the weights are smaller than or equal to 2, the optimum is attained for $f = 0$, so

$$\min_{f=0..y}\left\{x - y + 2f + \sum_{i=1}^{y-f} w(i)\right\} = x - y + \sum_{i=1}^{y} w(i). \qquad \square$$

Proposition 1. *Given an integer $x > 0$, the Euclidean distance values from $(0, 0)$ to the set $\{(x, y) \in Z^2, 0 \leq y \leq x\}$ is given without errors by the weight sequence $\left(1 + \sqrt{x^2 + i^2} - \sqrt{x^2 + (i-1)^2}\right)_{i=1..x}$.*

Proof. All weights in the weight sequence are smaller than 2 and the sequence is increasing, so by Lemma 1

$$\begin{aligned}
d\left(\mathbf{0},(x,y);W\right) &= x - y + \sum_{i=1}^{y} w(i)\\
&= x - y + \sum_{i=1}^{y}\left(1 + \sqrt{x^2 + i^2} - \sqrt{x^2 + (i-1)^2}\right)\\
&= x - y + \sqrt{x^2 + y^2} - (x - y)\\
&= d_E(\mathbf{0},(x,y)). \qquad \square
\end{aligned}$$

The following remark follows by the construction of the index set I in the previous section.

Remark 1. Proposition 1 holds for any permutation of the weights in the weight sequence.

Given the number of weights in the weight set, Proposition 1 and Remark 1 gives a set of weights that optimally approximates the Euclidean distance on the border of a square. Now, a reasonable order of the weights are computed in order to obtain the weight sequence. This is done by using a greedy algorithm, Algorithm 1.

Algorithm 1. Algorithm for computing suboptimal weight sequence from the sequence obtained by Proposition 1.

```
Input: A sequence of weights $1 \leq w(i) < 2, i = 1, .., k$, obtained by Proposition 1.
Output: A sequence $w'$ with suboptimal order of the weights.
Let $\mathcal{K} = \{1, 2, \ldots, k\}$ and $w' = (\infty, \infty, \ldots)$;
foreach $i = 1..k$ do
    $j' = \arg\min_{j \in \mathcal{K}} \left( \sum_{l=1..i} \left| d_E(i,l) - d(i,l;w'') \right|, \text{ where } w''(m) = \begin{cases} w'(m) \text{ for } m < i \\ w(j) \text{ for } m = i \\ \infty \text{ for } m > i \end{cases} \right)$;
    $w'(i) \leftarrow w(j')$;
    $\mathcal{K} \leftarrow \mathcal{K} \setminus \{j'\}$;
```

Given a square centered in $(0, 0)$ (a chessboard disk of radius k), for $x = 0..k$, the weight that minimizes the difference to the Euclidean distance in the next step is added to the weight sequence. The weight sets obtained by Proposition 1, for increasing x, are listed in Table 1.

Table 1. Optimal weight sequences (with rounded weights) obtained by Proposition 1. The first column shows the number of weights obtained by Proposition 1.

#	w(1)	w(2)	w(3)	w(4)	w(5)
1	1.4142				
2	1.2361	1.5924			
3	1.1623	1.4433	1.6371		
4	1.1231	1.3490	1.5279	1.6569	
5	1.0990	1.2861	1.4458	1.5722	1.6679

In the (greedy) Algorithm 1, the mean absolute difference between $d_E(\cdot, \cdot)$ and $d(\cdot, \cdot; W)$ is minimized in each step up to a radius 50. The sequences of weights obtained are listed in Table 2.

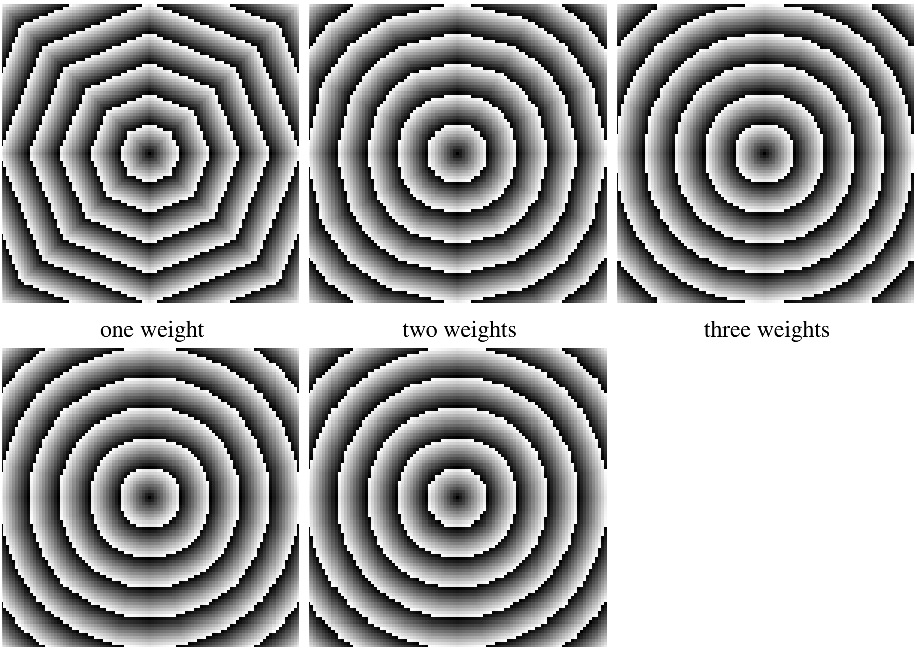
Table 2. Suboptimal weight sequences obtained by Algorithm 1 using the weights in Table 1. The sequences show the first 20 indices of the corresponding weight sequence in Table 1. The first column shows the number of weights in the sequence obtained by Proposition 1.

#	weight sequence
1	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
2	1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 1, 2, 1
3	2, 2, 1, 3, 1, 3, 2, 1, 3, 2, 1, 3, 2, 1, 3, 1, 3, 2, 1, 3
4	2, 3, 2, 1, 4, 3, 2, 1, 4, 2, 3, 1, 4, 2, 3, 1, 4, 2, 3, 1
5	3, 2, 4, 2, 3, 1, 5, 3, 2, 4, 1, 5, 3, 2, 4, 1, 5, 3, 2, 4

4 Distance Transform (DT)

The DT is a mapping from the image domain, a subset of the grid, to the range of the distance function. In a DT, each object grid point is given the distance from the closest background grid point. A modified version of a wave-front propagation algorithm can be used.

Now the formal definition of image is given.



one weight

two weights

three weights

four weights

five weights

Fig. 1. Distance maps from a single point using the weights and sequences in Table 1 and Table 2. The distance values are shown modulo 10.

Definition 1. *The image domain is a finite subset of $\mathbb{Z}^2$ denoted by $\mathcal{I}$. We call the function $F : \mathcal{I} \longrightarrow \mathcal{R}_d$ an image, where $\mathcal{R}_d$ is the range of the distance function d.*

An *object* is a subset $X \subset \mathcal{I}$ and the *background* is $\overline{X} = \mathcal{I} \setminus X$. We assume that $X, \overline{X} \neq \emptyset$. We denote the distance map for path-based distances with DM_d, where the subscript d indicates what distance function is used.

Definition 2. *The* distance map DM_d *generated by the distance function* $d(\cdot, \cdot; W)$ *of an object* $X \subset \mathcal{I}$ *is the mapping*

$$DM_d : \mathcal{I} \to \mathcal{R}_d \textit{ defined by}$$

$$P \mapsto d\left(\overline{X}, P; W\right), \textit{ where}$$

$$d\left(\overline{X}, P; W\right) = \min_{Q \in \overline{X}} \left\{ d\left(Q, P; W\right) \right\}.$$

In the case of W-distances with two weights, a minor modification of the Dijkstra algorithm (and with the same time complexity) can be used, see [17] and Theorem 4.1 in [13]. However, for multiple number of weights, this is not necessarily true. For W-distances, the used weights are also of importance, and they are determined by the *number of steps* of the minimal cost-path (not the cost). Therefore, in the general case of multiple weights presented here, we need to store this value also when propagating distance information. We define the auxiliary transform DM_s that holds the number of steps of the minimal cost path at each point, see Algorithm 2.

Note that we need to store not only the best distance values at the points, but the best values that are computed by various number of steps. Therefore for each point P a set $S(P)$ of pairs of values of the form (DM_d, DM_s) are stored with pairwise different DM_s. After the run of the algorithm for each point the minimal DM_d gives the result.

The novel Algorithm 2 shows how the distance map can be computed based on an extended optimal search (Dijkstra algorithm) and using the data structure described above. At the initialization the object points are the only points that are reached and it is done by 0 steps and with 0 cost. Then the border points of the object are listed in an increasing order by the minimal cost path already known for them. Actaully every point in the list has 0 cost, but the list will be updated by involving other points to where paths are already found. The **while** loop chooses (one of the) point(s) with minimal cost from the list since it is sure that we have the minimal cost path to this point already. Then in the loop the data of all neighbor points of the chosen point are updated by computing the cost of the new paths through the chosen point (having last step from the chosen point to the actual neighbor point). Therefore the algorithm holds the optimal distance attained at each point (as the usual algorithm), but this is done *for each* path length. So, if there are paths of different lengths ending up at the same point, distance information for each of the different path lengths are stored.

Algorithm 2. Computing DM for W-distances given by a weight sequence W.

```
Input: $W$ and an object $X \subset \mathbb{Z}^2$.
Output: The distance map $DM_d$.
Initialization: Let $S(P) \leftarrow \{(0,0)\}$ for grid points $P \in \overline{X}$. Let
$DM_d(P) = \min\{DM_d \mid (DM_d, DM_s) \in S(P)\}$. For all grid points $P \in X$ adjacent
to $\overline{X}$: push $(P, DM_d(P))$ to the list $L$ of ordered pairs sorted by increasing $DM_d(P)$.
while $L$ is not empty do
    Pop $(P, DM_d(P))$ from $L$;
    foreach $Q$: $Q, P$ are strict $2$-neighbors do
        foreach pair $(DM_d, DM_s) \in S(P)$ do
            if $w(DM_s + 1) \leq 2$ then
                if there is an element $(DM'_d, DM_s + 1) \in S(Q)$ then
                    if $DM'_d > DM_d + w(DM_s + 1)$ then
                        Replace $(DM'_d, DM_s + 1)$ by
                        $(DM_d + w(DM_s + 1), DM_s + 1)$ in $S(Q)$
                    end
                end
                else
                    Add $(DM_d + w(DM_s + 1),\ DM_s + 1)$ to $S(Q)$
                end
            end
        end
        Let $DM_d(Q) = \min\{DM'_d \mid (DM'_d, DM'_s) \in S(Q)\}$
        Push $(Q, DM_d(Q))$ to the ordered list $L$;
    end
    foreach $Q$: $Q, P$ are $1$-neighbors do
        foreach pair $(DM_d, DM_s) \in S(P)$ do
            if there is an element $(DM'_d, DM_s + 1) \in S(Q)$ then
                if $DM'_d > DM_d + 1$ then
                    Replace $(DM'_d, DM_s + 1)$ by $(DM_d + 1, DM_s + 1)$ in $S(Q)$
                end
            end
            else
                Add $(DM_d + 1, DM_s + 1)$ to $S(Q)$
            end
        end
        Push $(Q, DM_d(Q))$ to $L$
    end
end
```

5 Conclusions

When the optimal parameters are used, the distance function we have presented has very low rotational dependency. With our method one uses digital (path-based) distance that approximate the Euclidean distance on the grid points with small error. Still, the distance is defined as the minimal cost-path and can therefore be used, for example, to compute the distance map in an efficient way. We believe that the proposed distance function is potentially useful in many other image processing algorithms, for example for computing skeletons, or other algorithms where the low rotational independency is required.

References

1. Borgefors, G.: Distance transformations in digital images. Computer Vision, Graphics, and Image Processing 34, 344–371 (1986)
2. Celebi, M.E., Celiker, F., Kingravi, H.A.: On Euclidean norm approximations. Pattern Recognition 44(2), 278–283 (2011)
3. Das, P.P., Chakrabarti, P.P.: Distance functions in digital geometry. Information Sciences 42, 113–136 (1987)
4. Denev, A.: Digital distance functions defined by sequence of weights, Bachelor Thesis, Dept. of Information Technology, Uppsala University (2011)
5. Fabbri, R., da, F., Costa, L., Torelli, J.C., Bruno, O.M.: 2D Euclidean distance transform algorithms: A comparative survey. ACM Computing Surveys 40(1), 1–44 (2008)
6. Klette, R., Rosenfeld, A.: Digital geometry - geometric methods for digital picture analysis. Morgan Kaufmann (2004)
7. Mukherjee, J.: Hyperspheres of weighted distances in arbitrary dimension. Pattern Recognition Letters 34(2), 117–123 (2013)
8. Nagy, B.: Distance functions based on neighbourhood sequences. Publ. Math. Debrecen 63(3), 483–493 (2003)
9. Nagy, B.: Metric and non-metric distances on $\mathbb{Z}^n$ by generalized neighbourhood sequences. In: IEEE Proceedings of 4th International Symposium on Image and Signal Processing and Analysis (ISPA 2005), Zagreb, Croatia, pp. 215–220 (2005)
10. Nagy, B.: Distance with generalized neighbourhood sequences in nD and ∞D. Discrete Applied Mathematics 156(12), 2344–2351 (2008)
11. Ragnemalm, I.: The Euclidean distance transform in arbitrary dimensions. Pattern Recognition Letters 14(11), 883–888 (1993)
12. Rosenfeld, A., Pfaltz, J.L.: Sequential operations in digital picture processing. Journal of the ACM 13(4), 471–494 (1966)
13. Strand, R.: Distance Functions and Image Processing on Point-Lattices: with focus on the 3D face- and body-centered cubic grids. Ph.D. thesis, Uppsala University, Sweden (2008), `http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-9312`
14. Strand, R.: Weighted distances based on neighbourhood sequences. Pattern Recognition Letters 28(15), 2029–2036 (2007)
15. Strand, R., Nagy, B.: A weighted neighborhood sequence distance function with three local steps. In: IEEE Proceedings of 8th International Symposium on Image and Signal Processing and Analysis (ISPA 2011), Dubrovnik, Croatia, pp. 564–568 (2011)
16. Strand, R., Nagy, B., Borgefors, G.: Digital distance functions on three-dimensional grids. Theoretical Computer Science 412, 1350–1363 (2011)
17. Strand, R., Normand, N.: Distance transform computation for digital distance functions. Theoretical Computer Science 448, 80–93 (2012)
18. Yamashita, M., Ibaraki, T.: Distances defined by neighborhood sequences. Pattern Recognition 19(3), 237–246 (1986)

The Laplace-Beltrami Operator: A Ubiquitous Tool for Image and Shape Processing

Aaron Wetzler, Yonathan Aflalo, Anastasia Dubrovina, and Ron Kimmel

Technion - Israel Institute of Technology,
Haifa, Israel
{twerd,aflalo,nastyad,ron}@cs.technion.ac.il

Abstract. The ubiquity of the Laplace-Beltrami operator in shape analysis can be seen by observing the wide variety of applications where it has been found to be useful. Here we demonstrate a small subset of such uses with their latest developments including a scale invariant transform for general triangulated meshes, an effective and efficient method for denoising meshes using Beltrami flows via high dimensional embeddings of 2D manifolds and finally the possibility of viewing the framework of geodesic active contours as a surface minimization having the Laplace-Beltrami operator as its main ingredient.

Keywords: Laplace-Beltrami, denoising, scale invariant, active contours, segmentation.

1 Introduction

The Laplace-Beltrami operator is a generalization of the Laplacian to non-flat Riemannian manifolds. The Laplacian operator appears in differential equations describing various physical phenomena, such as heat diffusion, wave propagation, etc. In computer vision it has been used extensively, for example for blob and edge detection, or image smoothing. When working with cruved manifolds, such as 3D shapes or even images represented as surfaces in 3D or 4D, like volume images, MRI or CT, we need a general representation of the Laplacian operator, such that it will take into account the non-trivial geometry of the manifold. There are many instances in the field of shape analysis where various types of metrics play an important role. The common approach presented in this article is to view the problem within a manifold processing framework, and use an appropriately defined metric in order to calculate the Laplace-Beltrami. Both heat diffusion over a surface as well as minimal surfaces are direct applications of the operator. More advanced methods involving this ubiquitous operator include the generation of various shape descriptors, diffusion distance definition, isometry invariant embedding, to name just a few. In this paper we analyze several such methods from the point of view of manifold processing and illustrate their inherent inter-connectivity.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 302–316, 2013.

2 Beltrami Patch Denoising

The Laplace-Beltrami is the generalization of the Laplacian on Riemanian manifolds. It has been shown [25] that image denoising can be effectively formulated by considering an image to be a 2D manifold embedded into a higher dimensional spatial-spectral space such as $\{x, y, R, G, B\}$. This embedding can then be traversed iteratively using the so called Beltrami flow which generates a scale space over the manifold and leads to noise reduction of the image while preserving relevant features such as edges.

More recent papers [23][28] have extended this approach by embedding the image manifold into a so called patch-space. The improved denoising capacity of these techniques, as well as their greater generality, suggests that the Beltrami flow is indeed well suited for dealing with the problem of depth map and image denoising. An example of the smoothing property of the Beltrami patch flow can be seen in Fig. 1.

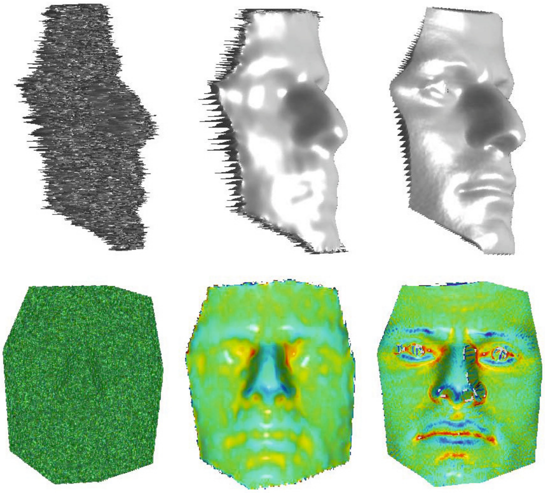


Fig. 1. Top row: Face with artificial Gaussian noise $\sigma = 10$, denoised using Beltrami patch denoising $PSNR = 42.45$, original face. Bottom row: Respective mean curvature of each face. It appears that despite the disruptive noise, the denoising process retains the main features present in the face.

2.1 Mathematical Background

We consider a height field I to be a $2D$ Riemannian manifold embedded in a higher dimensional space. We thus define the patch-space mapping $P : \Sigma \rightarrow M \subseteq \mathbb{R}^{n(2w+1)^2+2}$ such that

$$P\left(x, y\right) = \left(x, y, \left\{I^k\left(x+i, y+j\right)\right\}\right) \quad , \tag{1}$$

for $i, j = -w, .., w, \quad k = 1, .., n$ where $w \in \mathbb{N}$ is the window size and n is the number of channels we use. For the case of a single height field $n = 1$, however if we were provided with a set of registered scans of a particular surface, n could represent the number of scans. The manifolds Σ and M are equipped with metrics G and H respectively. We require that lengths between corresponding points measured on each manifold are the same. For that end, we write that

$$ds^2 = \left(dx\; dy\; dI^k_{i,j}\right) H \begin{pmatrix} dx \\ dy \\ dI^k_{i,j} \end{pmatrix} = (dx\; dy)\, G \begin{pmatrix} dx \\ dy \end{pmatrix}. \tag{2}$$

where $I^k_{i,j}$ is the compact form for $\left\{I^k\left(x + iw, y + jw\right)\right\}$. In reality, the coordinates x and y do not possess the same physical measure as the intensity values of the height field so we need to introduce a scaling factor into the patch-space metric given by

$$h_{pq} = \begin{cases} \delta_{pq} & 1 \le p, q \le 2 \\ \beta^2 \delta_{pq} & 2 < p, q \le n\left(2w + 1\right)^2 + 2 \end{cases}, \tag{3}$$

where δ_{pq} is the Kronecker delta. We can now use the chain rule $dI^k_{i,j} = \beta^2 I^k_{i,j\,x}$ $dx + \beta^2 I^k_{i,j\,y} dy$ from which it follows that when we pullback the metric from the embedding the induced metric tensor is given by

$$G = \begin{pmatrix} 1 + \beta^2 \sum_{i,j,k} I^{k\,2}_{i,j\,x} & \beta^2 \sum_{i,j,k} I^k_{i,j\,x} I^k_{i,j\,y} \\ \beta^2 \sum_{i,j,k} I^k_{i,j\,x} I^k_{i,j\,y} & 1 + \beta^2 \sum_{i,j,k} I^{k\,2}_{i,j\,y} \end{pmatrix}. \tag{4}$$

Using this metric we define a measure S on the manifold. For a Euclidean embedding in M, S is none other than the area of the surface as measured in Σ

$$S\left[\Sigma, G\right] = Area \propto \iint \sqrt{\det(G)}\, dx dy, \tag{5}$$

where the proportion is up to a scale as a result of the non-unity coefficients of the diagonal entries of H. There is a more general version of the above measure called the Polyakov action which can be useful for non-Euclidean embeddings and details of its application to the Beltrami framework can be found in [25]. We minimize Eq. (5) using variational calculus and then multiply by $g^{-1/2}$ which is permitted through the freedom of paramaterization. We arrive at

$$\Delta_g I^k_{i,j} = \frac{1}{\sqrt{g}} \mathrm{div}\left(\sqrt{g} G^{-1} \nabla I^k_{i,j}\right) = 0. \tag{6}$$

The left hand operator is recognized to be the Laplace-Beltrami operator and we can now compactly write the reformulation of the Beltrami flow in pach space as

$$I_t = \Delta_g I. \tag{7}$$

2.2 Method and Results

The Beltrami flow in patch-space derived above can produce efficient and plausible denoising schemes in natural images. The flow is performed by a novel explicit update scheme. We fix the number of iterations and find the optimal β for a given σ in order to maximize the quantative efficacy of the process as measured by the Peak Signal to Noise Ratio $PSNR = 10\log_{10}\left(1/E\left[(I_{est} - I)^2\right]\right)$ across a wide range of different representative images. σ is the noise level assumed to be present in the image. It has been empirically observed that these two variables have a surprizingly simple relationship for natural images as seen in Fig. 2.

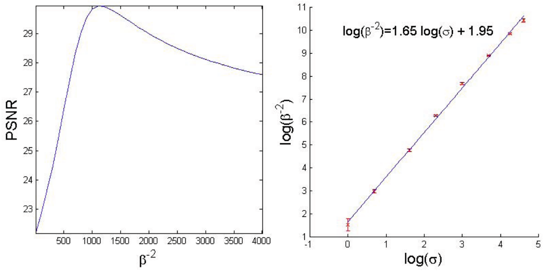


Fig. 2. Left: Example of $PSNR$ as a function of β^{-2} with a typical global maximum. Right: Linear relationship between the logarithms of σ and β^{-2}. Error bars indicate one standard deviation from the mean over a set of different images.

Accurate denoising of range images is often the first stage in 3D reconstruction pipelines that stitch depth images together to form a solid model and it is thus imperative that this process be fast and efficient. In our naive Matlab implementation we use an integral image and unit weighting of neighbouring height values to speed up the calculation of the flow update for each iteration. Furthermore, the diffusion update of every pixel in an image is independent of every other for a given iteration in our explicit scheme. This implies that an optimized implementation would need to take advantage of this inherrent parallelism and significant speed improvements could be expected. For implementation details of the method itself see [28].

Point clouds and meshes are often assumed to have noise which is Gaussian and that is offset along the normal direction to the true surface at every vertex location. In contrast, the noise model for range scanners is mainly observed to be offset in the direction of the capturing device's focal point. Furthermore the noise is not Gaussian in nature and also includes areas with missing data. This presents a significant hurdle for state of the art image denoising algorithms due to the fact that they are mostly tuned for optimal removal of additive white noise. A case in point is BM3D [13] to which we compare Beltrami patch denoising on a depth map obtained from a real scanner in Fig. 3.

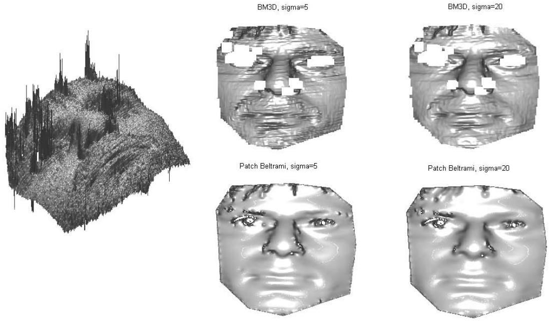


Fig. 3. Far left: Original noisy scan. Top row: BM3D for $\sigma = 5$ and $\sigma = 20$. Bottom row: Beltrami patch for $\sigma = 5$ and $\sigma = 20$.

Together with the efficiency of our method the Beltrami patch flow has an additional desirable property in that it can be tuned to eliminate weak high frequency structures. This is especially relevant for a variety of structured light scanners which tend to inject artifacts along the boundaries of the projected light patterns of the scanner. These artifacts can be observed as horizontal wrinkles in Fig. 3 and are faithfully preserved by BM3D whereas our method removes the artifacts while leaving the main components of the scan intact.

In this section we have illustrated the applicability of the Beltrami patch flow to height fields. One extension of this approach which is currently being investigated is how to apply the technique onto general meshes and point clouds. These data structures require a somewhat different treatment and a variety of new techniques are currently being developed by researchers to better understand and manipulate these discretized manifolds. With this in mind we now turn our attention to a new metric for use on triangulated meshes which is invariant to local changes in scale.

3 Invariant Decomposition of the Laplace-Beltrami Operator

In [26] it was argued that the Laplace-Beltrami operator and the corresponding heat operator acquire invariant properties when choosing an appropriately invariant metric. This observation was later exploited to construct an equi-affine invariant Laplace-Beltrami operator in [21] from which an invariant diffusion distance was extracted, and more recently in [1] where a scale invariant metric for surfaces was introduced.

At the other end, the standard Laplace-Beltrami operator's eigen-functions were used as a natural basis for shape representation, analogous to the Fourier basis in classical signal processing [16,17]. We follow [21] and show how the results of [1] can be incorporated into the axiomatic world of shape processing.

One of the useful properties of the Laplace-Beltrami is that the operator admits a spectral decomposition. In this section we show how a new metric coupled with a discrete version of the Laplace-Beltrami and its eigenfunctions can be used to produce locally scale invariant signatures of shapes.

3.1 Introduction

Consider $S(u, v)$ a parametrized surface $S : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$. The length of a parametrized curve C in S can be measured using a general parametrization p with

$$\begin{aligned} l(C) &= \int_{C \in S} ds = \int_C |C_p| dp = \int_C |S_u u_p + S_v v_p| dp \\ &= \int_C \sqrt{|S_u|^2 du^2 + 2\langle S_u, S_v\rangle du dv + |S_v|^2 dv^2}, \end{aligned} \tag{8}$$

from which we have the usual metric definition of infinitesimal distances on a surface

$$ds^2 = g_{ij} d\omega^i d\omega^j, \tag{9}$$

where we used Einstein summation convention, $\omega^1 = u, \omega^2 = v$ and $g_{ij} = \langle S_{\omega^i}, S_{\omega^j}\rangle$.

Using this convention and given a metric G on a manifold, we write the Laplace-Beltrami operator as

$$\Delta_G f = \frac{1}{\sqrt{g}} \partial_i \left(\sqrt{g} g^{ij} \partial_j f \right), \tag{10}$$

where g is the determinant of G, $g^{ij} = \left(G^{-1}\right)_{i,j}$, and ∂_i is the derivative with respect to the i^{th} coordinate.

The spectral theorem applied to the operator Δ_g states that it admits a spectral decomposition, i.e. an orthogonal eigenbasis $\{\phi_i, i \in \mathbb{Z}\}$ and a set of eigenvalues $\{\lambda_i, i \in \mathbb{Z}\}$ where

$$\Delta_g \phi_i = \lambda_i \phi_i, \quad \langle \phi_i, \phi_i \rangle = 1, \quad \forall i \in \mathbb{Z}. \tag{11}$$

This spectral decomposition has been extensively used for shape analysis. Diffusion geometry was introduced in [2] and refined in [12]. It uses the Laplace-Beltrami operator Δ_g of the surface as a diffusion or heat operator. The heat profile on the surface from a source located at s, after heat has dissipated for time t, is given by the heat kernel signature (HKS)[27,6]

$$\text{HKS}(s, t) = \sum_i e^{-\lambda_i t} \phi_i^2(s) \tag{12}$$

where ϕ_i and λ_i are the corresponding eigenfunctions and eigenvalues of Δ_g, that satisfy $\Delta_g \phi_i = \lambda_i \phi_i$. The diffusion distance is then defined as

$$\begin{aligned} d^2_{g,t}(s,s') &= \|h_{s,t}(\hat{s}) - h_{s',t}(\hat{s})\|^2_g \\ &= \int_S (h_{s,t}(\hat{s}) - h_{s',t}(\hat{s}))^2 da(\hat{s}) \\ &= \sum_i e^{-2\lambda_i t}(\phi_i(s) - \phi_i(s'))^2. \end{aligned} \tag{13}$$

The choice of an appropriate metric that stays invariant to certain classes of deformations is important in the context of shape analysis. Several distances have been used, such as Euclidean [11,3], geodesic [14,19,15,18], diffusion [8], and affine invariant versions thereof [22] to compare and match between shapes. Yet another example is the scale invariant *HKS* [9] which is a non-linear function of the *HKS*. In order to be robust to local scale changes and isometries the following local scale-invariant isometric metric was proposed

$$\tilde{g}_{ij} = |K|\langle S_{\omega^i}, S_{\omega^j}\rangle, \tag{14}$$

so that

$$d\tau^2 = |K| \left(\langle S_u, S_u\rangle du^2 + 2\langle S_u, S_v\rangle dudv + \langle S_v, S_v\rangle dv^2\right). \tag{15}$$

It is a similarity (local scale + isometry) invariant arc-length, where K is the gaussian curvature.

3.2 Experimental Results

To demonstrate the robustness of the scale invariant metric with respect to local scale transformation, we first consider the shapes shown in Fig. 4. We then present the profile of the heat kernel signature as a function of time (see Eq. (12)) at three different points, the left hand finger tip, the right hand one, and

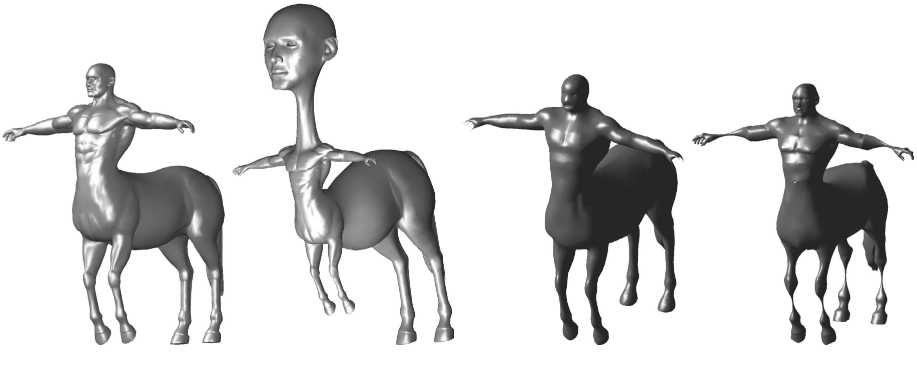


Fig. 4. Left to right: A centaur and its local scale transformed version. Details preservation for the coordinates reconstructed from the 1000 first eigenvectors of the Laplace Beltrami decomposition for the regular metric and the scale invariant metric.

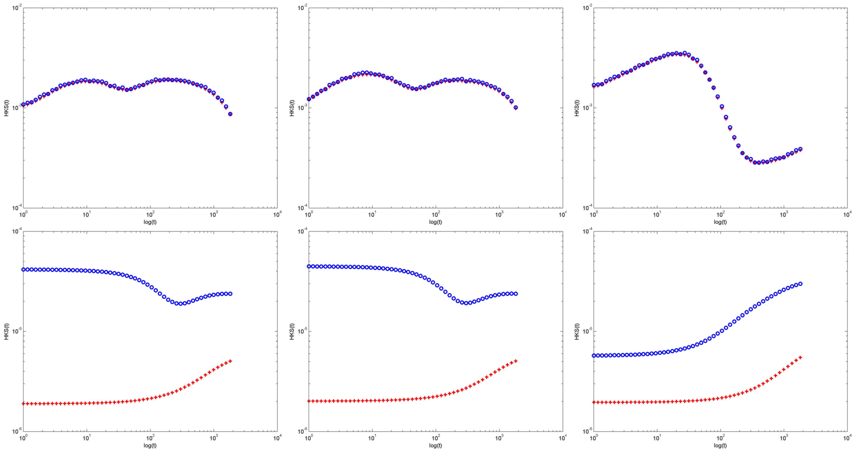


Fig. 5. Scaled heat kernel signatures for the regular metric (bottom), and the invariant version (top). The blue circles represent the signatures for three points on the original surface, while the red plus signs are computed from the deformed version. Using a log-log axes we plot the scaled-HKS as a function of t.

the horseshoe of the front left leg as shown in Fig. 5. Another experiment that allows us to understand the importance of an invariant metric is done by the following eigen reconstruction of a shape. The eigenbasis of the shape has been studied in [17] and [16]. We compute the spectral decomposition of the Laplace-Beltrami with respect to both usual and scale invariant metric and compute an approximation of the original shape by projecting the coordinates of the shape

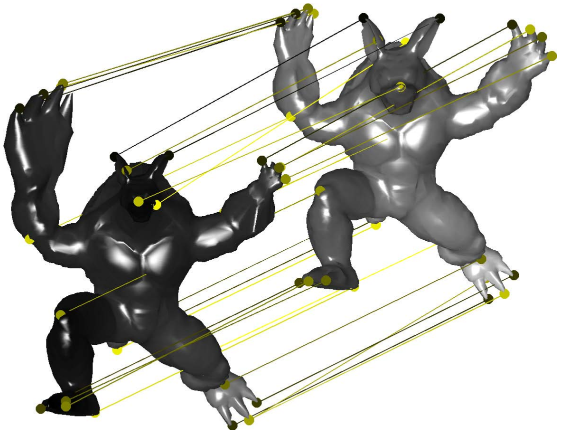


Fig. 6. Correspondences between the Armadillo and its local scale tranformed version using GMDS with the scale invariant metric

on the first one thousand eigenvectors in that basis. We notice that the local scale invariant Laplace-Beltrami has a spectral decomposition that preserves details at all scales, and can be useful not only to analyze a group of shapes with local scale transfomations but also shapes whose small details are important as shown in Fig. 4. In the last experiment depicted in Fig. 6 we applied the GMDS algorithm [7] using local scale invariant diffusion distance to compute the correspondence between an armadillo and its scaled version.

The Laplace-Beltrami's ubiquity can be seen not only in its application to shape understanding but also to the field of computer vision where its tendency towards minimal surfaces can be elegantly linked with the classic problem of segmentation through the geodesic active contours model.

4 Geodesic Active Contours and the Laplace-Beltrami Framework

In this section the Laplace-Beltrami operator and the flow towards minimal surfaces that it generates is shown to be intimately related to the Geodesic Active Contours (GAC) model for image segmentation [10]. Related work include the papers by Bresson *et al.* [5], who showed how to obtain the GAC flow by minimizing a *weighted Polyakov action*, and Bogdanova *et al.* [4], who showed similar results for embedding spaces other than 3-D Euclidean space. Sochen *et al.* [24] examined the image filtering problem, and showed that there existed an intimate relationship between the PDE-based geometric approaches, derived from minimizing the Polyakov action with an appropriate metric, and non-linear filters of robust statistics.

We proceed by briefly describing the two frameworks - the geodesic active contours and the generalized minimal surface flow, and then provide a formulation in which their relationship becomes evident.

4.1 Geodesic Active Contours

Geodesic active contours were introduced for object boundary detection using active contours evolving in time according to the intrinsic measures of the image. In [10] it was shown that curve evolution using the geodesic active contours model is equivalent to finding minimal distance curves in a Riemannian space whose metric is defined by the intrinsic image measures mentioned above. Specifically, these are curves that minimize the following length measure

$$L_R := \int f\left(\left|\nabla I\left(C(q)\right)\right|\right)\left|C'(q)\right| dq = \int f\left(\left|\nabla I\left(C(s)\right)\right|\right) ds, \tag{16}$$

in a Riemannian space with the metric tensor $g_{ij} = f\left(\left|\nabla I\left(C\right)\right|\right)\delta_{ij}$. The function $f\left(\left|\nabla I\left(C\right)\right|\right)$ is an edge indicator function, designed to stop the active contour when it arrives to the object boundary.

Casseles *et al.* used steepest-descent method to minimize the length L_R (16), and obtained the following curve evolution flow

$$\frac{\partial C(t)}{\partial t} = \left(f(I)\kappa - \nabla f \cdot \boldsymbol{N}\right)\boldsymbol{N}, \tag{17}$$

where κ is the curvature of C, and $\boldsymbol{N}$ is its unit inward facing normal.

The level set formulation [20] of the geodesic problem (17) is given by

$$\begin{aligned}\frac{\partial u}{\partial t} &= |\nabla u| \operatorname{div}\left(f(I)\frac{\nabla u}{|\nabla u|}\right) \\ &= f(I)\,|\nabla u| \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \nabla f(I) \cdot \nabla u, \end{aligned} \tag{18}$$

where $u(x, y, t)$ is the level set function of the evolving curve - $C(x, y, t) = \{(x, y) : u(x, y, t) = 0\}$, and $\kappa = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$ gives the curvature of the level sets of the function u.

The level set function flow in Eq. (18) can be interpreted as a generalized minimal surface flow. In order to show this we will use the methodology described by Sochen *et al.* in [25], which we will review briefly in the following section.

4.2 Generalized Minimal Surface Flow

Let us treat the level set function $u(x, y)$ as a two-dimensional surface embedded in a 3-dimensional space. As in Sec. 2 we define such an embedding by the map $\mathbf{X} : \mathit{\Sigma} \to M$, where $\mathit{\Sigma}$ denotes a 2-D manifold $\mathit{\Sigma}$ with local coordinates (σ_1, σ_2), and M denotes the 3-D embedding space M. Explicitly, $\mathbf{X}$ is written as $\mathbf{X} = \left(X^1(\sigma^1, \sigma^2), X^2(\sigma^1, \sigma^2), X^3(\sigma^1, \sigma^2)\right)$. Both manifolds $\mathit{\Sigma}$ and M are equipped with metric tensors, $g_{\mu\nu}(\sigma^1, \sigma^2)$ and $h_{ij}(x^1, x^2, x^3)$, respectively. The map $\mathbf{X}$ and the metric h_{ij} can be used to construct the metric on $\mathit{\Sigma}$

$$g_{\mu\nu}(\sigma^1, \sigma^2) = h_{ij}(\mathbf{X})\partial_\mu X^i \partial_\nu X^j. \tag{19}$$

Here we keep the Einstein summation convention, following the original work of [25]. Next, the following weight functional can be associated with the map $\mathbf{X} : \mathit{\Sigma} \to M$

$$S\left[X^i, g_{\mu\nu}, h_{ij}\right] = \int d^m\sigma \sqrt(g) g^{\mu\nu} \partial_\mu X^i \partial_\nu X^j h_{ij}(\mathbf{X}), \tag{20}$$

where $g^{\mu\nu}$ is the inverse of the metric $g_{\mu\nu}$ (that is $g^{\mu\gamma}g_{\gamma\nu} = \delta_{\mu\nu}$), and g is the determinant of $\{g_{\mu\nu}\}$. This weight functional is called Polyakov action, and can also be viewed as a generalized area measure. A particular case of the Polyakov action has already been shown in Sec. 2, Eq. (5).

The minimal weight map (embedding) $\mathbf{X}$ can be obtained using steepest-descent. The gradient of the Polyakov action with respect to the embedding is

$$-\frac{1}{2\sqrt{g}} h^{il} \frac{\delta S}{\delta X^l} = \frac{1}{\sqrt{g}} \partial_\mu \left(\sqrt{g} g^{\mu\nu} \partial_\nu X^i\right) + \mathit{\Gamma}^i_{jk} \partial_\mu X^j \partial_\nu X^k g^{\mu\nu} \tag{21}$$

In order to find the minimal measure embedding, Sochen *et al.* [25] used the following gradient descent flow

$$X_t^i = -\frac{1}{2\sqrt{g}} h^{il} \frac{\delta S}{\delta X^l}. \tag{22}$$

We note that the gradient (21) was obtained by multiplying the Euler-Lagrange equations of (20) by a strictly positive function and a positive definite matrix, that together were called a $pre-factor$. It does not change the minimum, and produces a geometric parameterization-invariant flow. We will see that the pre-factor needed to produce the GAC flow will be somewhat different, stemming from the different geometry of the problem. We can also see that the first term in the right-hand side of the Eq. (21), $\frac{1}{\sqrt{g}}\partial_\mu\left(\sqrt{g}g^{\mu\nu}\partial_\nu X^i\right)$ is exactly the Laplace-Beltrami operator acting on the embedding X, denoted above by $\varDelta_g$. The second term includes the Levi-Civita connection coefficients $\varGamma^i_{jk}$, that describe the geometry of the embedding space. When $M = \mathbb{R}^3$ with Euclidean metric, $h_{ij} = \delta_{ij}$, the second term vanishes, and the flow becomes $X_t = \varDelta_g X$, as seen in the previous sections.

4.3 Back to GAC: Level Set Formulation as a Flow Toward Minimal Surface

Next, we show that the level set geodesic active contour flow in Eq. (18) can be obtained by minimizing a certain generalized area measure. First, let us choose X that maps a 2D Euclidean space ($\sigma^1 = x, \sigma^2 = y$) to a 3D Euclidean space, such that

$$\mathbf{X} = (x, y, u(x, y)). \tag{23}$$

The action functional we would like to study is

$$S = \iint dxdy f(|\nabla I(x,y)|)\sqrt{1 + |\nabla u|^2}, \tag{24}$$

This is Polyakov action obtained by choosing the following metric tensors for the parameter and the embedding spaces $\varSigma$ and M, respectively

$$\begin{aligned} g_{\mu\nu} &= f(\sigma^2, \sigma^2)\left(\partial_\mu X \cdot \partial_\nu X\right), \\ h_{ij} &= f(x^1, x^2)\delta_{ij}. \end{aligned} \tag{25}$$

Both $g_{\mu\nu}$ and h_{ij} are legitimate metric tensors, and, since $(\sigma^2, \sigma^2) = (x, y)$ and $(x^1, x^2, x^3) = (x, y, z)$, Eq. (19) holds.

The metric tensor $g_{\mu\nu}$ written in a matrix form becomes

$$G = (g_{\mu\nu}) = f\begin{pmatrix} 1+u_x^2 & u_x u_y \\ u_x u_y & 1+u_y^2 \end{pmatrix} \tag{26}$$

The metric determinant is $g = \det(G) = f^2\left(1 + \|\nabla u\|^2\right)$, and $\sqrt{g} = f\sqrt{1 + \|\nabla u\|^2}$. The inverse of the metric is

$$G^{-1} = (g^{\mu\nu}) = \frac{f}{g}\begin{pmatrix} 1+u_y^2 & -u_x u_y \\ -u_x u_y & 1+u_x^2 \end{pmatrix}. \tag{27}$$

Next, we use Eq. (21) in order to obtain the gradient-descent flow for the level set function component of $\mathbf{X}$, namely $X^3 = u(x,y)$,

$$u_t = X^3_t = \frac{1}{\sqrt{g}}\partial_\mu\left(\sqrt{g}g^{\mu\nu}\partial_\nu u\right) + \mathit{\Gamma}^3_{jk}\partial_\mu X^j \partial_\nu X^k g^{\mu\nu}. \tag{28}$$

Let us develop the two terms of the flow in Eq. (28) separately. Substituting the expressions for $\sqrt{g}$ and $g^{\mu\nu}$ into the first term of the right-hand side of the flow in Eq. (28) produces

$$\frac{1}{\sqrt{g}}\partial_\mu\left(\sqrt{g}g^{\mu\nu}\partial_\nu u\right) = \frac{1}{f\sqrt{1+|\nabla u|^2}} \cdot \mathrm{div}\left(\frac{\nabla u}{\sqrt{1+|\nabla u|^2}}\right) \tag{29}$$

In order to calculate the second term of the flow in Eq. (28) we must find the expression for the Levi-Civita connection coefficients $\mathit{\Gamma}^3_{jk}$. For the metric h_{ij} defined in Eq. (25)

$$\begin{aligned}\mathit{\Gamma}^i_{jk} &= \frac{1}{2}h^{il}\left(\partial_j h_{lk} + \partial_k h_{jl} - \partial_l h_{jk}\right)\\ &= \sum_l \frac{1}{2}\frac{1}{f}\delta_{il}\left(\partial_j\left(f\delta_{lk}\right) + \partial_k\left(f\delta_{jl}\right) - \partial_l\left(f\delta_{jk}\right)\right)\\ &= \frac{1}{2}\frac{1}{f}\left(\delta_{ik}\partial_j f + \delta_{ji}\partial_k f - \delta_{jk}\partial_i f\right)\end{aligned} \tag{30}$$

Therefore

$$\mathit{\Gamma}^3_{jk} = \frac{1}{2f}\left(\delta_{3k}\partial_j f + \delta_{j3}\partial_k f\right), \tag{31}$$

or, in a matrix form,

$$\mathit{\Gamma}^3 = \frac{1}{2f}\begin{pmatrix} 0 & 0 & f_x \\ 0 & 0 & f_y \\ f_x & f_y & 0 \end{pmatrix}. \tag{32}$$

Finally, the second term of the flow from Eq. (28) becomes

$$\mathit{\Gamma}^3_{jk}\partial_\mu X^j \partial_\nu X^k g^{\mu\nu} = \frac{\nabla f \cdot \nabla u}{f^2\left(1+|\nabla u|^2\right)}. \tag{33}$$

Using Eq. (29) and Eq. (33) we obtain the expression for the level set function flow

$$u_t = \frac{1}{f\sqrt{1+|\nabla u|^2}}\mathrm{div}\left(\frac{\nabla u}{\sqrt{1+|\nabla u|^2}}\right) + \frac{\nabla f \cdot \nabla u}{f^2\left(1+|\nabla u|^2\right)} \tag{34}$$

In order to obtain the GAC level set formulation we need to multiply the above flow by a pre-factor of $f^2\left(1+|\nabla u|^2\right)$. The flow obtained this way is

$$u_t = \mathrm{div}\left(f\frac{\nabla u}{\sqrt{1+|\nabla u|^2}}\right)\sqrt{1+|\nabla u|^2} \tag{35}$$

We see that, up to the additional constant 1, this is exactly the flow of the level set function of the geodesic active contours model from Eq. (18). Since the surface definition in Eq. (23) is arbitrary, we can choose the aspect ratio between du and dx, dy to be as large as we want. Thus 1 can be viewed as ε that vanishes upon the right selection of u.

Essentially what we have shown is that the geodesic active contours method in its level set formulation can be regarded as a minimal surface detection problem that minimizes the Polyakov action functional, and in doing so we have determined the related metric tensors for both parameter and embedding spaces.

5 Conclusions

We have shown the links between a patch based heat flow with the Beltrami operator as a diffusion filter, an invariant metric that was introduced into the operator yielding invariant geometries for shape matching and synthesis, and the geodesic active contour model expressed as a Beltrami diffusion equation. The Laplace-Beltrami operator acting on data in one form or another can be seen as one of the most fundamental operators in the analysis and processing of images and shapes. By manipulating the metric, the action and the filtering processes we have observed its applicability across a wide range of problems which further illustrates the inherent ubiquity of the operator. We believe that the Laplace-Beltrami viewpoint in this field has the potential to enable and enhance the understanding and exploration of images and shapes.

Acknowledgments. This research was supported by the European Community's FP7-ERC program, grant agreement 267414.

References

1. Aflalo, Y., Raviv, D., Kimmel, R.: Scale invariant geometry for non-rigid shapes. Technical report, Technion University (2011)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. 15(6), 1373–1396 (2003)
3. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
4. Bogdanova, I., Bresson, X., Thiran, J.P., Vandergheynst, P.: Scale space analysis and active contours for omnidirectional images. IEEE Transactions on Image Processing 16(7), 1888–1901 (2007)
5. Bresson, X., Vandergheynst, P., Thiran, J.-P.: Multiscale active contours. In: Kimmel, R., Sochen, N.A., Weickert, J. (eds.) Scale-Space 2005. LNCS, vol. 3459, pp. 167–178. Springer, Heidelberg (2005)
6. Bronstein, A.M., Bronstein, M., Guibas, L.J., Ovsjanikov, M.: Shape google: Geometric words and expressions for invariant shape retrieval. ACM Transactions on Graphics 30(1), Article 1 (2011)

7. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. Proceedings of the National Academy of Science, pp. 1168–1172 (2006)
8. Bronstein, A.M., Bronstein, M.M., Kimmel, R., Mahmoudi, M., Sapiro, G.: A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. International Journal of Computer Vision 89(2-3), 266–286 (2010)
9. Bronstein, M., Kokkinos, I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In: Proc. Computer Vision and Pattern Recognition (CVPR), San Francisco, USA, December 13-18, pp. 1704–1711. IEEE Computer Society (2010)
10. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision 22(1), 61–79 (1997)
11. Chen, Y., Medioni, G.: Object modeling by registration of multiple range images. In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 3, pp. 2724–2729 (1991)
12. Coifman, R.R., Lafon, S.: Diffusion maps. Applied and Computational Harmonic Analysis 21(1), 5–30 (2006); Special Issue: Diffusion Maps and Wavelets
13. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on Image Processing, 2080–2095 (2007)
14. Elad (Elbaz), A., Kimmel, R.: On bending invariant signatures for surfaces. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 25(10), 1285–1295 (2003)
15. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L.: Topology matching for fully automatic similarity estimation of 3D shapes. In: ACM SIGGRAPH 2001, Los Angeles, CA, August 12-17 (2001)
16. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2000), pp. 279–286. ACM Press/Addison-Wesley Publishing Co., New York (2000)
17. Levy, B.: Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry. In: IEEE International Conference on Shape Modeling and Applications (SMI 2006), p. 13 (2006)
18. Mémoli, F., Sapiro, G.: A theoretical and computational framework for isometry invariant recognition of point cloud data. Foundations of Computational Mathematics 5(3), 313–347 (2005)
19. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Transactions on Graphics 21(4), 807–832 (2002)
20. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79(1), 12–49 (1988)
21. Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R., Sochen, N.: Affine-invariant geodesic geometry of deformable 3D shapes. In: Computers and Graphics, Herzliya, Israel, June 22-24. Proceedings of Shape Modelling International (SMI 2011), vol. 35(3). Elsevier (2011)
22. Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R., Sochen, N.: Affine-invariant geometry of deformable 3D shapes. In: Proc. of Computer vision and Pattern Recognition (CVPR). IEEE Computer Society (June 2011)

23. Roussos, A., Maragos, P.: Tensor-based image diffusions derived from generalizations of the total variation and Beltrami functionals. In: ICIP (September 2010)
24. Sochen, N., Kimmel, R., Bruckstein, A.M.: Diffusions and confusions in signal and image processing. Journal of Mathematical Imaging and Vision 14(3), 195–209 (2001)
25. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. IEEE Trans. on Image Processing, 310–318 (1998)
26. Sochen, N.A.: Stochastic processes in vision: From langevin to beltrami. In: IEEE International Conference on Computer Vision, vol. 1, p. 288 (2001)
27. Sun, J., Ovsjanikov, M., Guibas, L.J.: A concise and provably informative multiscale signature based on heat diffusion. Computer Graphics Forum 28(5), 1383–1392 (2009)
28. Wetzler, A., Kimmel, R.: Efficient beltrami flow in patch-space. In: Bruckstein, A.M., ter Haar Romeny, B.M., Bronstein, A.M., Bronstein, M.M. (eds.) SSVM 2011. LNCS, vol. 6667, pp. 134–143. Springer, Heidelberg (2012)

Towards Morphological Image Regularization Using the Counter-Harmonic Mean

Jorge Larrey-Ruiz[1], Rafael Verdú-Monedero[1],
Juan Morales-Sánchez[1], and Jesús Angulo[2]

[1] Dept. Tecnologías de la Información y las Comunicaciones,
Universidad Politécnica de Cartagena, 30202, Cartagena, Spain
jorge.larrey@upct.es
[2] CMM - Centre de Morphologie Mathématique, Mathématiques et Systémes,
MINES Paristech, 77300, Fontainebleau cedex, France

Abstract. The introduction of nonlinear filters which approximate flat dilation and erosion is an issue that has been studied during the past years. In the literature, we can find works which involve the definition of robust morphological-like filters from well-known operators such as the Counter-Harmonic Mean (CHM). The main goal of this paper is to provide the reader with a morphological CHM-based regularization which simultaneously preserve both the structural information in areas of the image with high gradient and the morphological effect in the areas with low gradient. With this purpose, we introduce a suitable mathematical framework and then deal with the variational formulation which is derived from it. Practical aspects of the implementation are discussed and some results are provided to illustrate the behaviour of our approach.

1 Introduction

Mathematical morphology operators [14] [16] are formulated in terms of geometric notions as well as in terms of complete lattice theory [11]. Morphological filters entail mainly the computation of supremum and infimum values in neighbourhoods (or structuring elements) which correspond respectively to the dilation and the erosion, the two basic operators. Morphological operators present also good scale-space properties [12] [4] but, by the natural duality of complete lattices, most operators appear by pairs, one acting on bright structures and the other one on dark structures. This latter property of asymmetry is in fact an advantage which allows defining evolved operators by product of a pair of dual ones. For instance, the opening (resp. closing) is obtained by the product of an erosion (resp. dilation) followed by a dilation (resp. erosion). The product of openings and closings leads to the alternate filters and other families of morphological filters [15] [16].

The idea of using the Counter-Harmonic Mean (CHM) [5] for constructing robust morphological-like operators, without the notions of supremum and infimum, was firstly proposed by van Vliet [17]. More recent works go further and exploit the CHM to introduce nonlinear filters whose effects mimic morphological dilation and erosion [2] [1]. In other words, CHM shows its appropriateness to approximate flat dilation and erosion. The general framework which results of these proposals can be utilized for

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 317–328, 2013.

the various algorithms of image diffusion. For example, as explained in [1], a pseudo-morphological counter-harmonic nonlinear diffusion can be defined from Perona and Malik-based models [13] [6]; comparing with respect to the standard filtering, the good properties of denoising without blurring are still present, and an additional effect of dilation/erosion is obtained.

In this work, the authors propose a specific regularization, within a variational framework, of the CHM, in order to improve the performance of the pseudo-morphological operators that can be derived from it. Our main aim is to come up with filters which simultaneously preserve not only the structural information in areas of the image with high gradient (such as edges or contours of objects) but also the pseudo-morphological effect in the areas with low gradient (such as textures). At this point, it should be clarified that our proposal has nothing to do with the total variation minimization with L^1 data fidelity introduced by Darbon [7], who obtained contrast invariant filters with a "morphological behaviour" which are particularly useful as a pre-processing stage before segmentation. The outline of the paper is as follows. We start out with the review of the notion of Counter-Harmonic Mean. In the following section, the proposed variational approach for the morphological image regularization (which is based on the CHM filter) is presented. Some interesting implementation aspects are also tackled in this section. In Section 4 our methodology is tested on three academic and actual experiments. Finally, the conclusions close the paper.

2 Counter-Harmonic Mean (CHM)

Let $\mathbf{a} = (a_1, a_2, \cdots, a_n)$ and $\mathbf{w} = (w_1, w_2, \cdots, w_n)$ be real n-tuples, i.e., $\mathbf{a}, \mathbf{w} \in \mathbb{R}^n$. Supposing that $r \in \bar{\mathbb{R}}$, the r-th Counter-Harmonic Mean (CHM) of $\mathbf{a}$ with weight $\mathbf{w}$ is given by [5]:

$$\mathfrak{K}^{[r]}(\mathbf{a};\mathbf{w}) := \begin{cases} \frac{\sum_{i=1}^{n} w_i a_i^r}{\sum_{i=1}^{n} w_i a_i^{r-1}}\,, & \text{if } r \in \mathbb{R} \\ \max(a_i)\,, & \text{if } r = +\infty \\ \min(a_i)\,, & \text{if } r = -\infty \end{cases} . \qquad (1)$$

The equal weight case will be denoted $\mathfrak{K}^{[r]}(\mathbf{a})$. We notice that $\mathfrak{K}^{[1]}(\mathbf{a};\mathbf{w})$ is the weighted arithmetic mean and $\mathfrak{K}^{[0]}(\mathbf{a};\mathbf{w})$ is the weighted harmonic mean.

The CHM has been considered in the literature of image processing as an appropriate filter to deal with salt and pepper noise [10]. More precisely, let $v = f(x,y)$ be a grey-level image: $f : \Omega \to \mathcal{V}$. Typically, for digital 2D images, $(x,y) \in \Omega$ where $\Omega \subset \mathbb{Z}^2$ is the discrete support of the image. The pixel values are $v \in \mathcal{V} \subset \mathbb{Z}$ or $\mathbb{R}$, but for the sake of simplicity of our study, we consider that $\mathcal{V} = [0,1]$. The CHM filter is obtained as

$$\kappa_B^P(f)(x,y) := \frac{\sum_{(l,m)\in B(x,y)} f(l,m)^{P+1}}{\sum_{(l,m)\in B(x,y)} f(l,m)^P} = \mathfrak{K}^{[P+1]}\left(\{f(l,m)\}_{(l,m)\in B(x,y)}\right) , \qquad (2)$$

where $B(x,y)$ is the window of the filter, centered at point (x,y), i.e., the region defined by the structuring element in the case of morphological operators. This filter is well suited for reducing the effect of pepper noise for $P > 0$ and of salt noise for $P < 0$.

In the pioneering paper [17], starting from the natural observation that morphological dilation and erosion are the limit cases of the CHM, i.e.,

$$\lim_{P \to +\infty} \kappa_B^P(f)(x,y) = \max_{(l,m) \in B(x,y)} (f(l,m)) = \delta_B(f)(x,y)\,, \tag{3}$$

and

$$\lim_{P \to -\infty} \kappa_B^P(f)(x,y) = \min_{(l,m) \in B(x,y)} (f(l,m)) = \varepsilon_B(f)(x,y)\,, \tag{4}$$

it was proposed to use the CHM to calculate robust nonlinear operators which approach the morphological ones but without using $\max$ and $\min$ operators. In addition, these operators are more robust to outliers (i.e., to noise) and consequently they can be considered as an alternative to rank-based filters in the implementation of pseudo-morphological operators.

It is easy to see that for $P \gg 0$ (resp. $P \ll 0$), the pixels with largest (resp. smallest) values in the local neighbourhood B will dominate the result of the weighted sum. Unlike the Mikowski mean [20], which does not reach the dilation (resp. erosion) even for $P = 100$ (resp. $P = -100$), the error in the results for CHM is already negligible for $P = 20$ (resp. $P = -20$). In the exhaustive study carried out in [1], it is also shown that the convergence to the erosion with $P \ll 0$ is faster than to the dilation with equivalent $P \gg 0$, i.e., assuming $P > 0$,

$$\left|\kappa_B^P(f)(x,y) - \delta_B(f)(x,y)\right| \geq \left|\kappa_B^{-P}(f)(x,y) - \varepsilon_B(f)(x,y)\right|\,, \quad \forall (x,y) \in \Omega\,.$$

Finally, it should be noted that $\kappa_B^P(f)$ and $\kappa_B^{-P}(f)$ are not dual operators with respect to the complement, i.e., for $P > 0$,

$$\kappa_B^P(f) \neq \complement \kappa_B^{-P}(\complement f)\,,$$

with $\complement f = 1 - f$. In particular, the following ordering relationship holds for $P > 0$:

$$\kappa_B^{-P}(f)(x,y) \leq \kappa_B^P(f)(x,y)\,.$$

Flat dilation, erosion, and derived filters, commute with any strictly increasing mapping A, i.e., $v_1 \leq v_2 \Leftrightarrow A(v_1) \leq A(v_2)$, $v_1, v_2 \in \mathcal{V}$; which means for instance that $\delta_B\left((A(f)\right) = A\left(\delta_B(f)\right)$. This property is related to the fact that morphological operators are increasing (or order preserving): for any pair of images f and g, if $f(x,y) \leq g(x,y)$ then $\delta_B(f)(x,y) \leq \delta_B(g)(x,y)$. CHM filter is generally not increasing; however, this is not limiting for the purpose of robust approximation to dilation and erosion.

As said above, we limit here our developments to positive bounded images f in $\mathcal{V} = [0,1]$. We note that, for the general case of $\mathcal{V} \subset \mathbb{R}$, we have $\kappa_B^P(f) = \complement \kappa_B^P(\complement f)$; that is, the CHM filter is self-dual with respect to the involution $\complement f = -f$, meaning that the CHM filter approximates a dilation for positive values and an erosion for negative values.

3 Regularization of the CHM-Based Filtering

Mathematically, the regularized gray-level image $u(x,y)$ which results from the application of our methodology to the original image $f(x,y)$ has to meet two requirements. On one hand, it has to be similar to the outcome of the CHM filter $\kappa_B^P(f)$ in areas of the image with low gradient (such as textures). And on the other hand, it has to preserve the structural information (i.e., edges or contours) in the areas with high gradient. This problem can be approached in terms of the variational calculus [8] [9], as explained in the following paragraphs.

The goal of the variational calculus is to obtain a function (i.e., an image, in the current scenario) that minimizes a certain energy functional (or cost function) $\mathcal{E}(u)$:

$$u := \arg\min_u \mathcal{E}(u)\,, \tag{5}$$

where, in a first approximation, the energy functional could be defined as

$$\mathcal{E}(u) := \frac{1}{2}\int_\Omega \left(u - \kappa_B^P(f)\right)^2 dxdy\,. \tag{6}$$

It should be noted that the latter definition leads to the trivial solution $u = \kappa_B^P(f)$, and therefore the pretended requirements are unmet. As an alternative, in this work we propose the following joint energy functional:

$$\mathcal{E}(u) := \frac{\alpha}{2}\int_\Omega \left(u - \kappa_B^P(f)\right)^2 dxdy + \frac{1}{2}\int_\Omega \|\nabla f\|^2 (u-f)^2 dxdy\,, \tag{7}$$

where $\alpha > 0$ is a scalar parameter, usually referred to as the regularization parameter, which is used to control and weight the influence of the first term of the functional (morphological effect) versus the second term (fitting to the data). In the second term, the squared difference between the regularized image and the original image is weighted by the squared magnitude of the gradient of the latter; hence the fitting term does not contribute to the cost function (7) in the spatial positions where the gradient is negligible, thus allowing for a noticeable pseudo-morphological filtering in these regions.

According to the variational calculus, a necessary condition for a minimizer u of the joint energy functional is that the first variation of $\mathcal{E}(u)$ in any direction (also known as the *Gâteaux* derivative) vanishes for all suitable perturbations $z \in \mathbb{R}^{\#\Omega}$, i.e., $d\mathcal{E}(u;z) = 0$. This yields the following expression:

$$d\mathcal{E}(u;z) = \int_\Omega \left\langle \alpha\left(u - \kappa_B^P(f)\right) + \|\nabla f\|^2 (u-f), z\right\rangle_{\mathbb{R}} dxdy = 0\,, \tag{8}$$

where $\langle\cdot,\cdot\rangle_{\mathbb{R}}$ denotes the dot (or inner) product in $\mathbb{R}$. The proof is detailed in the Appendix A. Since (8) must be valid for all z, we can conclude that

$$\alpha\left(u - \kappa_B^P(f)\right) + \|\nabla f\|^2 (u-f) = 0\,. \tag{9}$$

Equation (9) is the Euler-Lagrange equation which corresponds to the minimization problem (5); this expression models the equilibrium state that u has to achieve in order to be an actual minimizer of $\mathcal{E}(u)$.

With the purpose of solving the Euler-Lagrange equation, a time-marching scheme can be employed by adding an artificial time t and then computing the steady-state solution. This strategy gives rise to the equation

$$\frac{\partial}{\partial t}u + \alpha\left(u - \kappa_B^P(f)\right) + \|\nabla f\|^2(u - f) = 0\,, \tag{10}$$

where $u = u(x, y; t)$; in the steady-state, $\frac{\partial}{\partial t}u = 0$ and (10) holds (9). Equation (10) can be solved numerically by discretizing the time (i.e., $t := \xi\tau$, with $\xi \in \mathbb{N}$ being the iteration index and where $\tau > 0$ is the time-step), and then replacing the temporal partial derivative by its discrete approximation (first backward difference):

$$\frac{\partial}{\partial t}u(x, y; t) \approx \frac{u(x, y; \xi\tau) - u(x, y; \xi\tau - \tau)}{\tau}\,. \tag{11}$$

Using the notation $u^{(\xi)}(x, y) := u(x, y; \xi\tau)$, the resulting semi-implicit iteration for the regularized image is the following:

$$u^{(\xi)} = u^{(\xi-1)} + \tau\left(\alpha(\kappa_B^P(f) - u^{(\xi-1)}) + \|\nabla f\|^2(f - u^{(\xi-1)})\right)\,, \tag{12}$$

where $u^{(\xi)}$ is initialized to the CHM filter, i.e., $u^{(0)} := \kappa_B^P(f)$.

3.1 Practical Implementation of the Algorithm

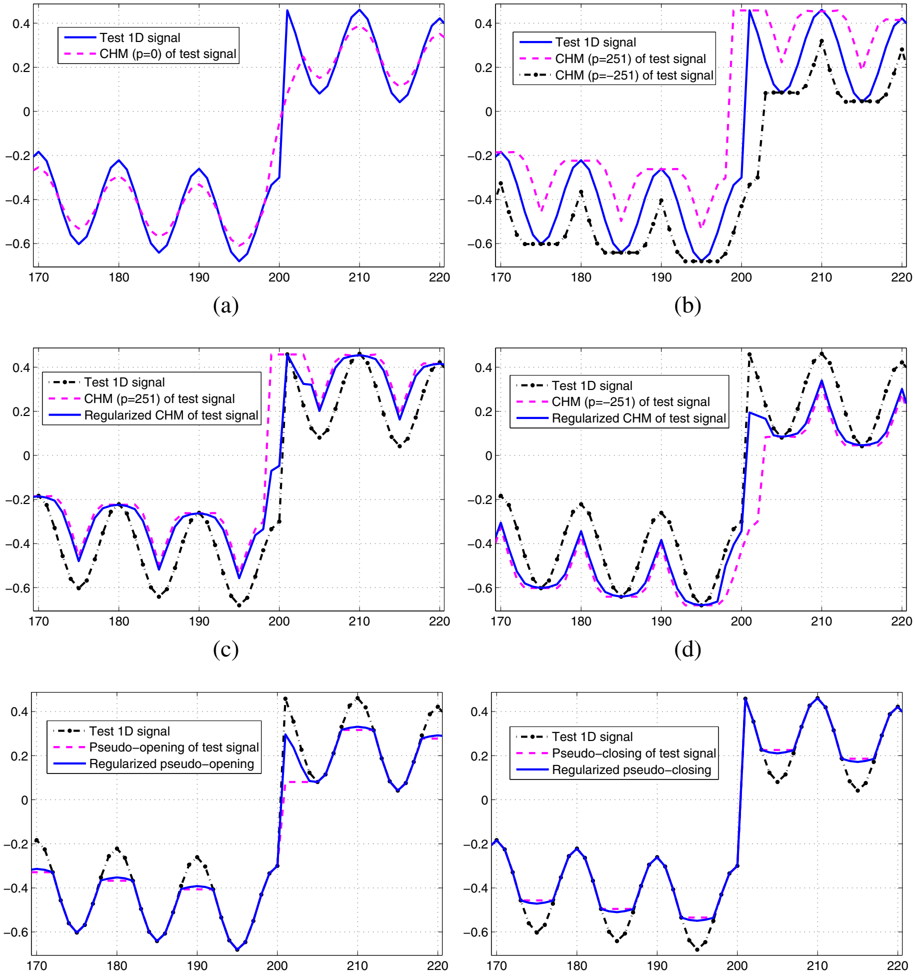
When tackling the software implementation of the proposed regularization scheme, some practical issues have to be considered. Firstly, the gray levels of the original image must be normalized (i.e., $f(x, y) \in [0, 1]$) prior to the computation of $\kappa_B^P(f)$. This way, overflow-related problems that could appear for high values of P when calculating the CHM (see (2)) are avoided. Once the outcome of this filter is computed, the values of both f and $\kappa_B^P(f)$ are restored to the original domain $\mathcal{V} \subset \mathbb{Z}$ or $\mathbb{R}$.

Another practical subject which deserves our attention is the following: in order to control the evolution of the iteration (12), the squared magnitude of the gradient is also normalized, i.e., $\|\nabla f\|^2 \in [0, 1]$. As a consequence, one can guess an approximate value for the regularization parameter α which is appropriate for a particular application scenario: if $\alpha \ll 1$, the weight of the fitting term is higher and then the regularized image u will be closer to the original image f; if $\alpha \approx 1$, the pseudo-morphological effect introduced by the CHM filter $\kappa_B^P(f)$ will be more noticeable in the regularized image, even in the regions with high gradient.

Lastly, with the purpose of guaranteeing stability, the chosen value for the time-step must satisfy $\tau < (\sum_{l=1}^{2} 2/h_l)^{-1}$, where h_l denotes the grid size in the l direction (see e.g. [19]). Working on a uniformly-spaced regular grid, we assume $h_l = 1$ for horizontal and vertical directions, hence $\tau < 0.25$.

4 Results

In this section, the proposed methodology is tested on three experiments involving data of different nature. In all cases, the quality of the procedure has to be appreciated subjectively by the reader (i.e., it has to be assessed by means of visual inspection). Additionally, we provide the computed value of the SSIM index [18], which measures the structural similarity between the original image and the filtered ones (where SSIM $= 1$ means a perfect match). In order to display the results, we use close-up views of the original images with the aim of showing clearly the most illustrative features of the outputs. For all experiments, we have chosen a time-step $\tau = 0.2$, a flat window $B(x, y)$ for the filter (i.e., an all-ones structuring element), and $\xi = 100$ iterations of the algorithm.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 1. Experiment 1: synthetic data. (a) Test signal and mean filter; (b) Pseudo-dilation and pseudo-erosion; (c) Pseudo-dilation vs regularized pseudo-dilation; (d) Pseudo-erosion vs regularized pseudo-erosion; (e) Pseudo-opening vs regularized pseudo-opening; (f) Pseudo-closing vs regularized pseudo-closing.

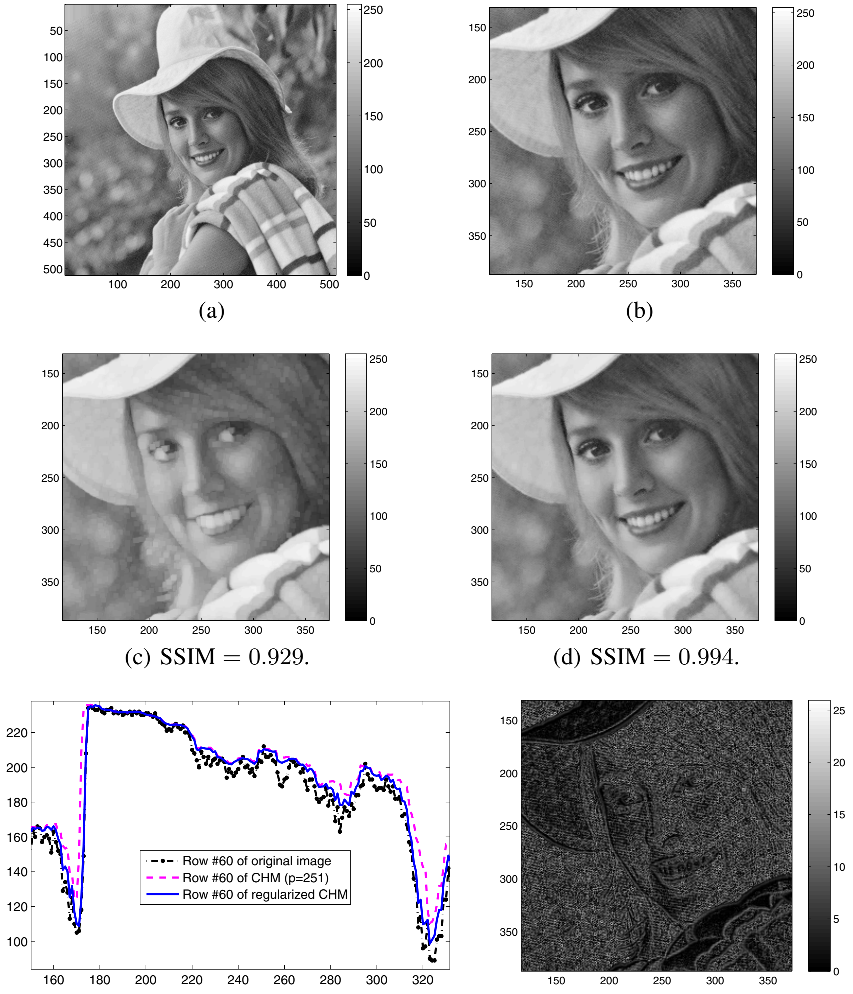
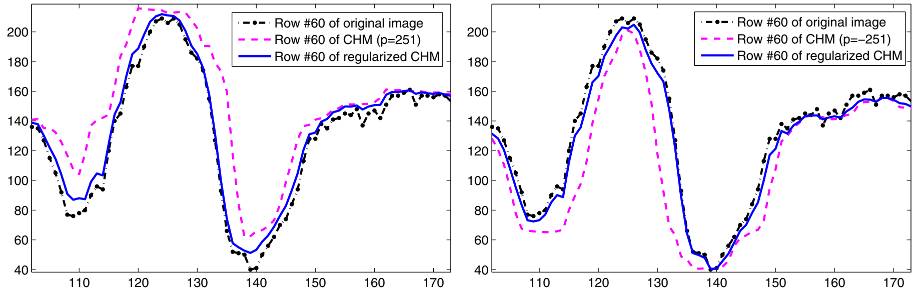


Fig. 2. Experiment 2: noisy picture. (a) Original image; (b) Zoom-in of the original image; (c) Pseudo-dilation; (d) Regularized pseudo-dilation; (e) Pseudo-dilation vs regularized pseudo-dilation (row #60); (f) Absolute difference between (b) and (d).

Simulated data is used to evaluate our method first. The input is a combination of sinusoidal 1D signals and the step function (see Fig.1). In Fig.1(a), it can be seen how the CHM with $P = 0$ equals a standard mean filtering. Fig.1(b) shows the pseudo-dilation and pseudo-erosion of the test signal; as expected, the pseudo-dilation is always over the test signal, whereas the pseudo-erosion is always under it. The order of the CHM filter P (± 251) is high (resp. low) enough for a good approximation to standard dilation (resp. erosion). The outcome of regularizing the pseudo-dilation (resp. pseudo-erosion) is shown in Fig.1(c) (resp. Fig.1(d)); it can be easily appreciated how the regularized signal is close to the test signal around the high gradient zone, while accurately approximates the corresponding pseudo-morphological operator in the regions with low gradient magnitude. Finally, Fig.1(e) and Fig.1(f) show respectively the non-regularized and regularized pseudo-opening (i.e., pseudo-erosion+pseudo-dilation+regularization) and pseudo-closing (i.e., pseudo-dilation+pseudo-erosion+regularization); it should be noted how in Fig.1(e) the regularized pseudo-opening follows the test signal better than the non-regularized pseudo-opening in the zone with high gradient. The value of the regularization parameter is $\alpha = 0.7$.

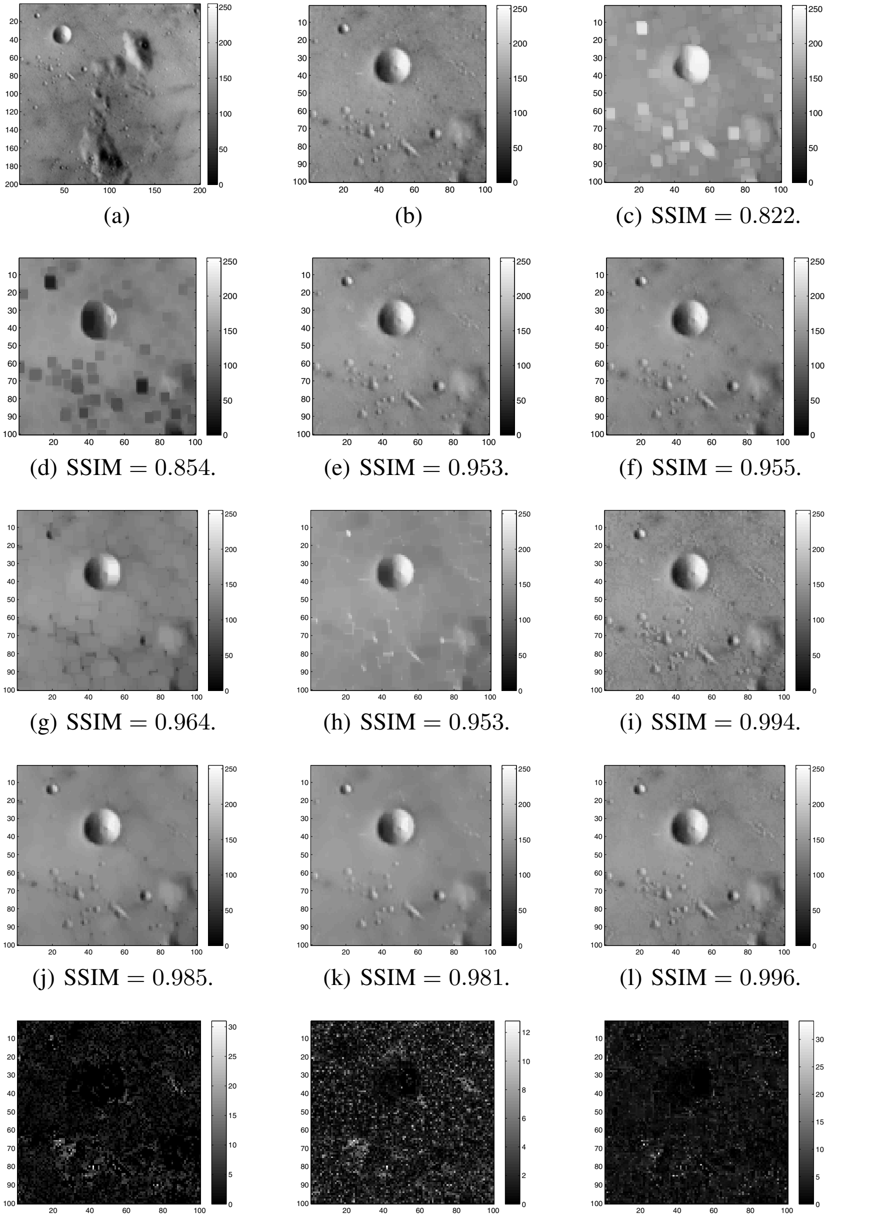
The second experiment focuses on a non-academic example. Fig.2 contains the results of applying the proposed framework to an actual gray-level image (a picture with low pepper-like noise), see Fig.2(a)-2(b). In Fig.2(c) and Fig.2(d) the pseudo-dilation and the regularized pseudo-dilation of the original image are respectively shown ($P = 251$, $\alpha = 0.02$). When comparing Fig.2(b) and Fig.2(d), it is noticed that the small dark fluctuations have been blurred while the edges (see eyes, lips, or stripes of the blanket) have been preserved (thus preventing an excessive morphological effect). This can also be appreciated in Fig.2(e), where corresponding rows of the non-regularized and the regularized pseudo-dilated images are displayed; the regularized "signal" follows either the original image if the gradient is high, or the output of the pseudo-morphological filter if the gradient is low. Finally, the absolute difference (which includes the removed pepper-like noise) between Fig.2(b) and Fig.2(d) can be seen in Fig.2(f).

To further demonstrate the performance of our algorithm over actual gray-level images, it is tested on a photograph of the surface of the moon (Fig.4(a)-4(b)). The standard morphological dilation and erosion, and the regularized ($\alpha = 0.2$) pseudo-dilation ($P = 251$) and pseudo-erosion ($P = -251$) are respectively shown in Fig.4(c)-4(f). Dark regions are pronounced in (pseudo-)erosion schemes whereas bright regions are emphasized in (pseudo-)dilation schemes, but it should be noted that the CHM-based regularized versions do not pay in exchange a loss of resolution (or, in other words, the structural information is preserved). Indeed, in Fig.3 it can be seen how the regularized pseudo-dilation and regularized pseudo-erosion behave as expected: they are close to the original image when the gradient magnitude is high, but also approximate accurately the corresponding pseudo-morphological filtering when the gradient magnitude is low. If we consider at this point the product of the previous operators, i.e.,



(a) (b)

Fig. 3. Experiment 3: surface of the moon. (a) Pseudo-dilation vs regularized pseudo-dilation (row #60); (b) Pseudo-erosion vs regularized pseudo-erosion (row #60).



(a)

(b)

(c) SSIM = 0.822.

(d) SSIM = 0.854.

(e) SSIM = 0.953.

(f) SSIM = 0.955.

(g) SSIM = 0.964.

(h) SSIM = 0.953.

(i) SSIM = 0.994.

(j) SSIM = 0.985.

(k) SSIM = 0.981.

(l) SSIM = 0.996.

(m)

(n)

(o)

Fig. 4. Experiment 3: surface of the moon. (a) Original image; (b) Zoom-in of the original image; (c) Standard dilation; (d) Standard erosion; (e) Regularized pseudo-dilation; (f) Regularized pseudo-erosion; (g) Standard opening; (h) Standard closing; (i) Standard toggle mapping; (j) Regularized pseudo-opening; (k) Regularized pseudo-closing; (l) Pseudo-toggle mapping; (m) Absolute difference between (b) and (i); (n) Absolute difference between (b) and (l); (o) Absolute difference between (i) and (l).

(pseudo-)opening and (pseudo-)closing, we can also study the differences between the standard morphological results and the proposed regularized outputs. Fig.4(g)-4(i) show respectively the standard opening, closing, and the so-called toggle mapping (constructed by choosing at each spatial position the value of the pixel from either the opening or the closing whose gray-level is closer to that of the original image); in particular, it is noticed that there exists an excessive "blurring" (or morphological effect) in the opening and the closing, and that the toggle mapping is even more noisy than the original image. Now let us focus on Fig.4(j)-4(l), where the regularized pseudo-opening, the regularized pseudo-closing and the corresponding pseudo-toggle mapping are respectively shown; when comparing these figures with Fig.4(b), it can be concluded that the regularized operators act as expected. Moreover, if we pay attention to the pseudo-toggle mapping, we can see that it is much closer to the original image than the standard toggle mapping (the exact differences are displayed in Fig.4(m)-4(o)), while being at the same time less noisy (i.e., it has an overall "morphological behaviour").

We have considered for the examples given in this paper large enough values of $|P|$ which guarantee a good approximation to the dilation (resp. erosion). In fact, the same regularization could be applied directly to the output of the dilation (resp. erosion). However, by taking smaller values, the approximation turns out to be robust against outliers (i.e., noise); this point will be developed in ongoing research.

5 Conclusions

In this work, a new methodology has been proposed for approaching the morphological regularization of an image. In order to meet this goal, we incorporate the Counter-Harmonic Mean (CHM) into a variational framework and then mathematically deduce the expressions which give rise to the regularized image. The suitable iteration is explicitly provided, as well as some practical issues which have to be considered when dealing with the software implementation of the algorithm. The fundamental property of the resulting scheme is its ability to preserve simultaneously both the structural information in areas of the image with high gradient magnitude and a pseudo-morphological effect in the regions with low gradient magnitude. The performance of the proposed methodology has been validated by means of illustrative examples involving simulated and actual data.

In ongoing research, we would like to explore alternative applications of the variational framework in order to address how to find the optimal order of the CHM filter, i.e., a value of P such as the resulting pseudo-morphologically regularized image provides the lowest possible energy of the joint functional to be minimized. We would also like to establish links between our theoretical approach (more precisely, the pseudo-toggle mapping) and the total variation (TV-L^1) minimization.

Acknowledgements. This work is supported by the Spanish Ministerio de Ciencia e Innovación, under grant TEC2009-12675.

A Computation of the Gâteaux Derivative of the Joint Energy Term

From (8) and the definition of the *Gâteaux* derivative (see e.g. [3]), the expression of $d\mathcal{E}(u;z)$ can be deduced as follows:

$$
\begin{aligned}
&d\mathcal{E}(u;z) = \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left(\mathcal{E}(u+\epsilon z) - \mathcal{E}(u)\right)\\
&= \lim_{\epsilon\to 0} \frac{1}{2\epsilon} \int_{\Omega} \alpha \left(u + \epsilon z - \kappa_B^P(f)\right)^2 + \|\nabla f\|^2 (u+\epsilon z - f)^2 - \alpha\left(u - \kappa_B^P(f)\right)^2 - \|\nabla f\|^2 (u-f)^2 dxdy\\
&= \lim_{\epsilon\to 0} \frac{1}{2\epsilon} \int_{\Omega} \alpha \left((u - \kappa_B^P(f))^2 + 2\epsilon(u-\kappa_B^P(f))z + (\epsilon z)^2\right) + \|\nabla f\|^2 \left((u-f)^2 + 2\epsilon(u-f)z + (\epsilon z)^2\right)\\
&\quad - \alpha\left(u - \kappa_B^P(f)\right)^2 - \|\nabla f\|^2 (u-f)^2 dxdy\\
&= \lim_{\epsilon\to 0} \int_{\Omega} \alpha \left((u - \kappa_B^P(f))z + \frac{\epsilon}{2} z^2\right) + \|\nabla f\|^2 \left((u-f)z + \frac{\epsilon}{2} z^2\right) dxdy\\
&= \int_{\Omega} \left\langle \alpha\left(u - \kappa_B^P(f)\right) + \|\nabla f\|^2 (u-f), z\right\rangle_{\mathbb{R}} dxdy = 0\,,
\end{aligned}
$$

where $z \in \mathbb{R}^{\#\Omega}$ is an arbitrary perturbation and $\langle\cdot,\cdot\rangle_{\mathbb{R}}$ denotes the dot (or inner) product in $\mathbb{R}$. This result is used in Section 3 in order to obtain the Euler-Lagrange equation (9) which describes the equilibrium state needed for the regularization of the CHM-based filtering.

References

1. Angulo, J.: Generalised morphological image diffusion. Submitted to Computer Vision and Image Understanding
2. 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)
3. Behmardi, D., Nayeri, E.D.: Introduction of Frećhet and Gâteaux derivative. Applied Mathematical Sciences 2(20), 975–980 (2008)
4. van den Boomgaard, R., Dorst, L.: The morphological equivalent of gaussian scale-space. In: Proc. of Gaussian Scale-Space Theory, pp. 203–220 (1997)
5. Bullen, P.S.: Handbook of Means and their Inequalities, 2nd edn. Springer (1987)
6. Cattle, 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)
7. Darbon, J.: Total variation minimization with l^1 data fidelity as a contrast invariant filter. In: 4th International Symposium on Image and Signal Processing (ISPA), pp. 221–226 (2005)
8. Forsyth, A.R.: Calculus of Variations. Dover Publishing (1960)
9. Gelfand, I.M., Fomin, S.V.: Calculus of Variations. Dover Publishing (2000)
10. González, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley, Boston (1992)
11. Heijmans, H.J.A.M.: Morphological Image Operators. Academic Press, Boston (1994)
12. Jackway, P.T., Deriche, M.: Scale-space properties of the multiscale morphological dilation-erosion. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(1), 38–51 (1996)
13. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)
14. Serra, J.: Image Analysis and Mathematical Morphology, vol. I. Academic Press, London (1982)

15. Serra, J.: Image Analysis and Mathematical Morphology, vol. II: Theoretical Advances. Academic Press, London (1988)
16. Soille, P.: Morphological Image Analysis. Springer (1999)
17. van Vliet, L.J.: Robust local max-min filters by normalized power-weighted filtering. In: Proc. IEEE International Conference of Pattern Recognition, vol. 17, pp. 696–699 (2004)
18. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)
19. Weickert, J.: Coherence-enhancing diffusion filtering. International Journal of Computer Vision 31(2-3), 111–127 (1999)
20. 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)

A Learning Framework for Morphological Operators Using Counter–Harmonic Mean*

Jonathan Masci[1], Jesús Angulo[2], and Jürgen Schmidhuber[1]

[1] IDSIA – USI – SUPSI, Manno–Lugano, Switzerland
jonathan@idsia.ch,
http://idsia/~masci/
[2] CMM-Centre de Morphologie Mathématique,
Mathématiques et Systèmes, MINES ParisTech, France

Abstract. We present a novel framework for learning morphological operators using counter-harmonic mean. It combines concepts from morphology and convolutional neural networks. A thorough experimental validation analyzes basic morphological operators dilation and erosion, opening and closing, as well as the much more complex top-hat transform, for which we report a real-world application from the steel industry. Using online learning and stochastic gradient descent, our system learns both the structuring element and the composition of operators. It scales well to large datasets and online settings.

Keywords: mathematical morphology, convolutional networks, online learning, machine learning.

1 Introduction

Mathematical morphology (MM) is a nonlinear image processing methodology based on computing max/min filters in local neighborhoods defined by structuring elements [16,17]. By concatenation of two basic operators, i.e., the dilation $\delta_B(f)$ and the erosion $\varepsilon_B(f)$, on the image f, we obtain the closing $\varphi_B(f) = \varepsilon_B(\delta_B(f))$ and the opening $\gamma_B(f) = \delta_B(\varepsilon_B(f))$, which are filters with scale-space properties and selective feature extraction skills according to the underlaying structuring element B. Other more sophisticated filters are obtained by combinations of openings and closings, to address problems such as non-Gaussian denoising, image regularization, etc.

Finding the proper pipeline of morphological operators and structuring elements in real applications is a cumbersome and time consuming task. In the machine learning community there has always been lot of interest in learning such operators, but due to the non-differentiable nature of the max/min filtering only few approaches have been found to succeed, notably one based on LMS (gradient steepest descent algorithm) for rank filters formulated with the sign function

[*] This work has been supported by ArcelorMittal, Maizières Research, Measurement and Control Dept., France.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 329–340, 2013.

[13,14]. This idea was later revisited [12] in a neural network framework combining morphological/rank filters and linear FIR filters. Other attempts from the evolutionary community (e.g., genetic algorithms [7] or simulated annealing [21]) use black-box optimizers to circumvent the differentiability issue. However, most of the proposed approaches do not cover all operators. More importantly, they cannot learn both the structuring element and the operator, e.g., [11]. This is obviously a quite important limitation as it makes very hard or even impossible the composition of complex filtering pipelines. Furthermore, such systems are usually limited to a very specific application and hardly generalize to complex scenarios.

Inspired by recent work [1] on counter-harmonic mean asymptotic morphology, we propose a novel framework to learn morphological pipelines of operators. We combine convolutional neural networks (CNN) with a new type of layer that permits complex pipelines through multiple layers and therefore extends this models to a Morphological Convolutional Neural Network (MCNN). It extends previous work on deep learning while making directly applicable all optimization tricks and findings of this field.

Here we focus on methodological foundations and show how the model learns several operator pipelines, from dilation/erosion to top-hat transform (i.e., residue of opening/closing). We report an important real-world application from the steel industry, and present a sample application to denoising where operator learning outperforms hand-crafted structuring elements.

Our main contributions are:

- a novel framework where learning of morphological operators and filtering pipelines can be performed using gradient-based techniques, exploiting recent insights of deep-learning approaches;
- the introduction of a novel pseudo-morphological layer for CNN, which we name PConv, to let CNN benefit from highly nonlinear, morphology-based filters;
- the stacking of many PConv layers, to learn complex pipelines of operators such as opening, closing and top-hats.

2 Background

Here we illustrate the foundations of our approach, introducing the Counter-Harmonic Mean formulation for asymptotic morphology and CNN.

2.1 Asymptotic Morphology Using Counter-Harmonic Mean

We start from the notion of counter-harmonic mean [2], initially used [19] for constructing robust morphological-like operators. More recently, its morphological asymptotic behavior was characterized [1]. Let $f(x)$ be a 2D real-valued image, i.e., $f : \Omega \subset \mathbb{Z}^2 \rightarrow \mathbb{R}$, where $x \in \Omega$ denotes the coordinates of the pixel in the image domain Ω. Given a (positive) weighting kernel $w : W \rightarrow \mathbb{R}_{+}$, W

being the support window of the filter, the *counter-harmonic mean (CHM) filter of order* P, $-\infty \leq P \leq \infty$ is defined by,

$$\kappa_w^P(f)(x) = \frac{(f^{P+1} * w)(x)}{(f^P * w)(x)} = \frac{\int_{y \in W(x)} f^{P+1}(y) w(x-y) dy}{\int_{y \in W(x)} f^P(y) w(x-y) dy}, \tag{1}$$

where f^P is the image, where each pixel value of f is raised to power P, / indicates pixel-wise division, and $W(y)$ is the support window of the filter w centered on point y. We note that the CHM filter can be interpreted as $P-$*deformed convolution*, i.e., $\kappa_w^P(f)(x) \equiv (f *_P w)(x)$. For $P \gg 0$ ($P \ll 0$) the pixels with largest (smallest) values in the local neighborhood W will dominate the result of the weighted sum (convolution), therefore morphological dilation and erosion are the limit cases of the CHM filter, i.e., $\lim_{P \to +\infty} (f *_P w)(x) = \sup_{y \in W(x)} f(y) = \delta_W(f)(x)$, and $\lim_{P \to -\infty} (f *_P w)(x) = \inf_{y \in W(x)} f(y) = \varepsilon_W(f)(x)$, where W plays the role of the structuring element. As proven earlier [1], apart from the limit cases (e.g., a typical order of magnitude of $5 \leq |P| < 10$), we have the following behavior:

$$(f *_P w)(x) \mid_{P \gg 0} \approx \sup_{y \in W(x)} \left\{ f(\mathbf{y}) + \frac{1}{P} \log \left(w(x-y) \right) \right\}, \tag{2}$$

$$(f *_P w)(x) \mid_{P \ll 0} \approx \inf_{y \in W(x)} \left\{ f(\mathbf{y}) - \frac{1}{P} \log \left(w(x-y) \right) \right\}, \tag{3}$$

which can be interpreted, respectively, as the nonflat dilation (supremal convolution) and nonflat erosion (infimal convolution) using the structuring function $b(x) = \frac{1}{P} \log \left(w(x) \right)$. By using constant weight kernels, i.e., $w(x) = 1$ if $x \in W$ and $w(x) = 0$ if $x \notin W$, and $|P| \gg 0$, we just recover the corresponding flat structuring element W, associated to the structuring function $\mathfrak{w}(x) = 0$ if $x \in W$ and $\mathfrak{w}(x) = -\infty$ if $x \notin W$.

From a precise morphological viewpoint, we notice that for finite P one cannot guarantee that $(f *_P w)(x)$ yields exactly a pair of dilation/erosion, in the sense of commutation with max/min [16,17]. Consequently, *stricto sensu*, we can only name them as pseudo-dilation ($P \gg 0$) and pseudo-erosion ($P \ll 0$). The asymptotic cases of the CHM filter can be also combined to approximate opening and closing operators, i.e.,

$$\begin{cases} \left((f *_{-P} w) *_P w \right)(x) \xrightarrow{P \gg 0} \gamma_W(f)(x), \\ \left((f *_P w) *_{-P} w \right)(x) \xrightarrow{P \ll 0} \varphi_W(f)(x). \end{cases} \tag{4}$$

2.2 Convolutional Neural Networks

CNN are hierarchical models alternating two basic operations, convolution and subsampling, reminiscent of simple and complex cells in the primary visual cortex [8]. Their main characteristic is that they exploit the 2D structure of images via weight sharing, learning a set of convolutional filters. Certain CNN scale

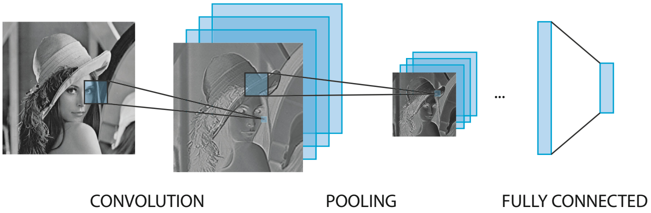


Fig. 1. A schematic representation of an CNN. Convolutional and pooling layers are stacked below the fully connected layers used for classification.

well to real-sized images and excel in many object recognition [3,4,6,10] and segmentation [5,18] benchmarks. We refer to a state-of-the-art CNN as depicted in Figure 1. It consists of several basic building blocks briefly explained here:

- *Convolutional Layer*: performs a 2D filtering between input images $\{f_i\}_{i \in I}$ and a bank of filters $\{w_k\}_{k \in K}$, producing another set of images $\{h_j\}_{j \in J}$ denoted as maps. Input-output correspondences are specified through a connection table CT (inputImage i, filterId k, outputImage j). Filter responses from inputs connected to the same output image are linearly combined. This layer performs the following mapping: $h_j(x) = \sum_{i,k \in CT_{i,k,j}} (f_i * w_k)(x)$, where $*$ indicates the 2D valid convolution. Then, a nonlinear activation function (e.g., *tanh*, *logistic*, etc.) is applied to h_j. Recently *relu* activations have been found to excel. They are the units we use for all our models. A *relu* unit operates as $relu(x) = \max(lb, \min(x, ub))$. It is common choice to use 0 as lower bound (lb) and ∞ as upper bound (ub).
- *Pooling Layer*: down-samples the input images by a constant factor keeping a value (e.g. maximum or average) for every non overlapping subregion of size p in the images. Max-Pooling is generally favorable, as it introduces small invariance to translation and distortion, leads to faster convergence and better generalization [15].
- *Fully Connected Layer*: this is the standard layer of a multi-layer network. It performs a linear multiplication of the input vector by a weight matrix.

Note the striking similarity between the max-pooling layer and a dilation transform. The former is in fact a special case of dilation, with a square structuring element of size p followed by downsampling (sampling one out of every p pixels). Our novel layer, however, does not any longer limit the pooling operation to simple squares, but allows for a much richer repertoire of structuring elements fine-tuned for given tasks. This is what makes MCNN so powerful.

3 Method

Now we are ready to introduce the novel morphological layer based on CHM filter formulation, referred to as **PConv** layer. For a single channel image $f(x)$

and a single filter $w(x)$ the PConv layer performs the following operation

$$PConv(f; w, P)(x) = \frac{(f^{P+1} * w)(x)}{(f^P * w)(x)} = (f *_P w)(x) \tag{5}$$

It is parametrized by P, a scalar which controls the type of operation ($P < 0$ pseudo-erosion, $P > 0$ pseudo-dilation and $P = 0$ standard linear convolution), and by the weighting kernel $w(x)$, where the corresponding asymptotic structuring function is given by $\mathfrak{w}(x) = \log(w(x))$. Since this formulation is differentiable we can use gradient descent on these parameters.

The gradient of such a layer is computed by back-propagation [20,9]. In minimizing a given objective function $L(\theta; X)$, where θ represents the set of parameters in the model, back-propagation applies the chain rule of derivatives to propagate the gradients down to the input layer, multiplying them by the Jacobian matrices of the traversed layers. Let us introduce first two partial results of back-propagation

$$\varDelta_U(x) = \frac{f(x)}{(f^P * w)(x)}; \quad \varDelta_D(x) = \frac{-f(x) \cdot (f^{P+1} * w)(x)}{(f^P * w)(x)}. \tag{6}$$

The gradient of a PConv layer is computed as follows

$$\frac{\partial L}{\partial w} = \tilde{f}^{P+1} * \varDelta_U + \tilde{f}^P * \varDelta_D \tag{7}$$

$$\frac{\partial L}{\partial P} = f^{P+1} \cdot \log(f) \cdot (\frac{f}{f^P * w} * \tilde{w}) + f^P \cdot \log(f) \cdot (\varDelta_D * \tilde{w}) \tag{8}$$

where $\tilde{f}, \tilde{w}$ indicate flipping along the two dimensions and $\cdot$ indicates elementwise multiplication. The partial derivative of the PConv layer with respect to its input (to back-propagate the gradient) is instead

$$\frac{\partial (f *_P w)(x)}{\partial f} = \varDelta_U(x) + \varDelta_D(x). \tag{9}$$

Learning the top-hat operator requires a *short-circuit* in the network to allow for subtracting the input image (or the result of an intermediate layer) from the output of a filtering chain. For this purpose we introduce the **AbsDiffLayer** which takes two layers as input and emits the absolute difference between them. Partial derivatives can still be be back-propagated.

3.1 Learning Algorithm

Minimizing a PConv layer is a non-convex, highly non-linear operation prone to local convergence. Deep-learning findings tell us that stochastic gradient descent is the most effective algorithm to train such complex models. In our experiments we use its full online version where weights are updated sample by sample. The learning rate decays during training. To further avoid bad local minima we use a momentum term. For the opening/closing tasks we also alternate between learning P, keeping w fixed, and vice-versa. This is common in online dictionary learning and sparse coding. We also constrain $w \geq 0$.

4 Experiments

We thoroughly evaluate our MCNN on several tasks. First we assess the quality of dilation/erosion operators, which require a single PConv layer. This gives a good measure of how well training can be performed using the CHM derivation. Then we investigate a two-layer network learning openings/closings. This is already a challenging task hardly covered in previous approaches.

We then learn the top-hat transform for a very challenging steel industry application. Using 2 filters per layer we learn to simultaneously detect two families of defects without resorting to multiple training. Our implementation allows for learning multiple filters for every layer, thus producing a very rich set of filtered maps. Subsequent convolutional layers can learn highly nonlinear embeddings. (We believe that this will also dramatically improve segmentation capabilities of such models.) We also show that a simple CNN does *not* learn well pipelines of morphological operators. This is actually expected *a priori* due to the nature of conventional convolutional layers, and shows the value of our novel PConv layer.

As final benchmark we consider denoising. Our MCNN shows the superiority of operator learning over hand-crafting structuring elements for non-Gaussian (binomial and salt-and-pepper) noise. We also show that our approach performs well on total variation (TV) approximation for additive Gaussian noise.

In all our experiments we use stochastic gradient descent and a filter size of 11×11 unless otherwise stated. The per-pixel mean-squared error loss (MSE) is used.

4.1 Learning Dilation and Erosion

In this first set of experiments we create a dataset as follows: for every input image f_i we produce a target image t_j using a predetermined flat structuring element B_k and a predetermined operator: $t_j = \delta_{B_k}(f_i)$ or $t_j = \varepsilon_{B_k}(f_i)$. We train until convergence. Overfitting is not an issue in such a scenario. The net actually learns the true function underlying the data. In fact, for an image of 512×512 and a structuring element with support of 11×11 there are 502^2 patches, way more than the 11^2 elements in the structuring element. A CNN with equal topology fails, producing mainly Gaussian blurred images, illustrating the need for a PConv layer to handle this kind of nonlinearities. Figure 2 shows the results of a dilation with three structuring elements: a line of 15 pixels and 45°, a square of size 5 pixels and a diamond of side 5 pixels. Figure 3 shows similar results for the erosion transform. Note that the learned weighted kernels $w(x)$ are not exactly uniformly equal to 1. The corresponding morphological structuring function $\mathfrak{w}(x)$, obtained after applying the logarithm on the weights, produces a rather flat shape. In practice, we observed that learning an erosion is slightly more difficult than learning the dual dilation. This is related to the asymmetric numerical behavior of CHM for $P > 0$ and $P < 0$. Nevertheless, in all cases the learned operator has excellent performance.

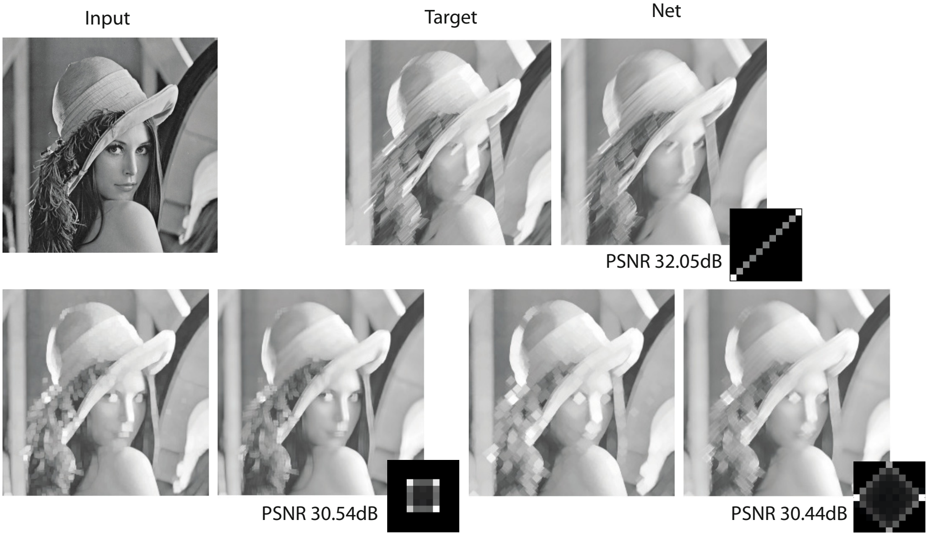


Fig. 2. Examples of learning a dilation with three different structuring elements. The target and net output are slightly smaller than the original image due to valid convolution. The obtained kernel $w(x)$ for each case is also depicted.

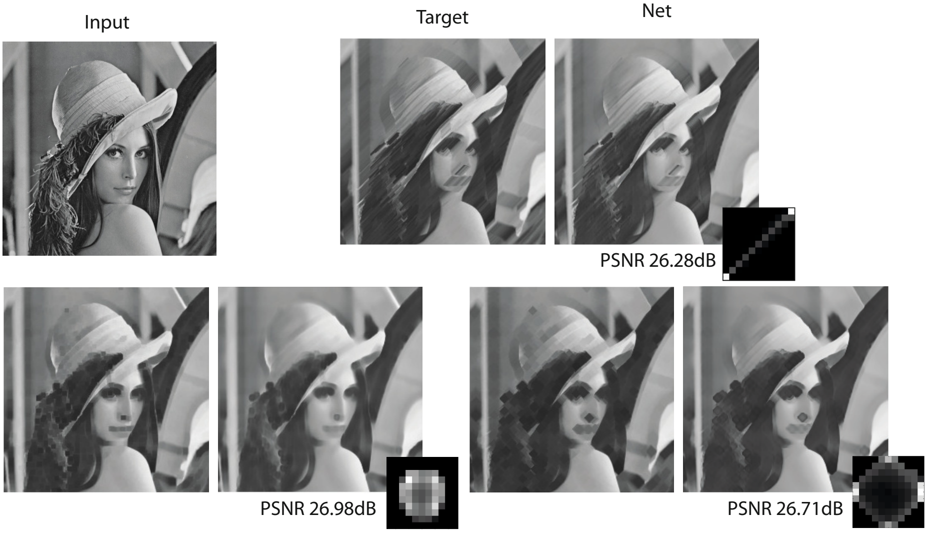


Fig. 3. Examples of learning an erosion with three different structuring elements along with the learned kernel $w(x)$

4.2 Learning Opening and Closing

In this set of experiments we train our system to learn openings $\gamma_{B_k}(f_i)$ and closings $\varphi_{B_k}(f_i)$. Learning such functions is extremely difficult. To the best of our knowledge, we are the first to do this in a flexible and gradient-based framework without any prior. For instance, in classical approaches [13] or more recent ones [11], the operator needs to be fixed *a priori*.

Figure 4-top shows an example of a closing with a line of length 10 and an orientation of 45°, whereas Figure 4-bottom shows an example of an opening with a square of size 5. In both cases, the obtained kernel for the first $L1$ and second $L2$ PConv layers are depicted. We see that the associated structuring element is learned with a good approximation. On the other hand, however, we also start to see that learning a flat opening/closing is remarkably hard and that the network output starts to be slightly "blurry". On the one hand this is due by the obtained values of P, e.g., in the closing $P_{L1} = 6.80$ and $P_{L2} = -8.85$, in the opening $P_{L1} = -7.64$ and $P_{L2} = 7.07$ are in the interval of asymptotically unflat behavior. On the other hand, they are not totally symmetric. We intend to further study these issues in ongoing work.

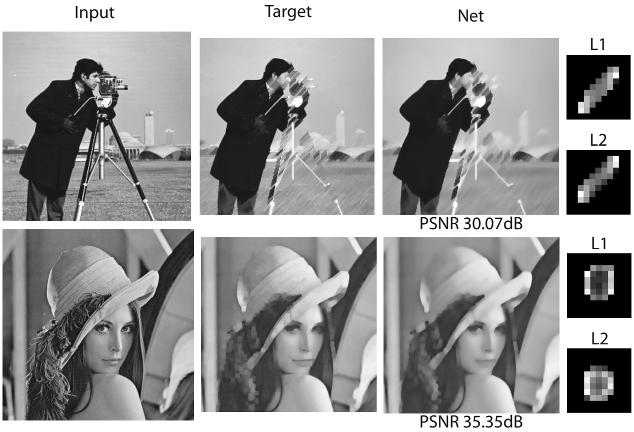


Fig. 4. *Top*: an example of learning a closing operator where a line of length 10 and orientation 45° is used. *Bottom*: opening with a square structuring element of size 5. The network closely matches the output and almost perfectly learns the structuring elements in both PConv layers.

4.3 Learning Top-Hat Transform

Delegating the learning to a neural network allows for easily constructing complex topologies by linking several simple modules. We recall that the white top-hat is the residue of the opening, i.e., $\varrho^{+}_{B_k}(f_i) = f_i - \gamma_{B_k}(f_i)$, and the black top-that the residue of the closing, i.e., $\varrho^{-}_{B_k}(f_i) = \varphi_{B_k}(f_i) - f_i$. Thus, to learn a

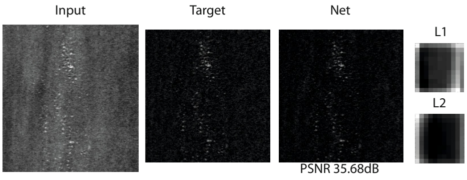


Fig. 5. Learning a top-hat transform. The defected input image has bright spots to be detected. The network performs almost perfectly on this challenging task.

top-hat transform we introduce the **AbsDiff** layer. It takes two layers as input and emits their absolute difference in output. Backpropagation is performed as usual.

Top-hat is particularly relevant in real applications such as steel surface quality control. It is a powerful tool for defect detection. Here we first show that our framework can learn such a transform. Increasing the number of filters per layer from 1 to 2, we show that our MCNN is also much more powerful when jointly learning two transforms. Figure 5 shows the results for a single top-hat. We create our training set by applying a white top-hat ϱ_B^+, where B is a disk of size 5 pixels. This operator extracts only the structures of size smaller than B and brighter than the background. We clearly see that the network performs almost perfectly. To further assess the advantages of a PConv layer over a conventional convolutional layer, we also train a CNN with identical topology. The discrepancy between the two models in terms of losses (MSE) is large: we have 1.28E-3 for our MCNN and 1.90E-3 for the CNN. More parameters are required for a CNN to reach better performance. This clearly establishes the added value of our MCNN.

Defect detection in steel industry requires many top-hat operations to extract bright/dark structures, which can greatly vary in size and shape. Tuning them one by one is a cumbersome process. Furthermore, several models' outputs need to be considered to obtain the final detection result. Figure 6 shows that by simply increasing the number of filters per layer we can simultaneously learn two top-hat transforms and address this problem. We learn a white top-hat with a disk of size 5 and a black top-hat with a line of size 10 and orientation of 0°. A convolutional layer is used to combine the output of the two operators. The architecture is as follows: 2 PConv layers, Conv layer, AbsDiff layer with the input. The network is almost perfect from our viewpoint. This opens the possibility of using such a setup in more complex scenarios where several morphological operators should be combined. This is of great interest in multiple class steel defect detection.

4.4 Learning Denoising and Image Regularization

Finally we compare our MCNN to conventional morphological pipelines in the denoising task. Morphological filters are recommended for non-Gaussian denoising. The purpose of this evaluation, however, is not to propose a novel noise

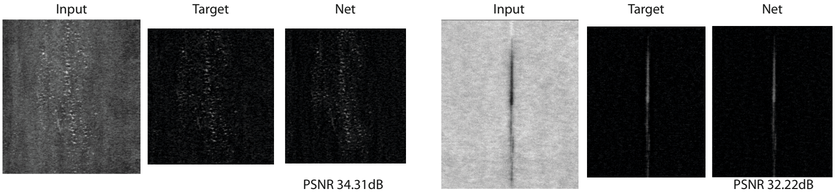


Fig. 6. Learning two top-hat transforms. On the left, bright spots need to be detected. On the right, a dark vertical line. The network performs almost perfectly in this challenging task.

removal approach, but to show the advantages of a learnable pipeline over a hand-crafted one. We start with *binomial noise* where 10% of the image pixels are switched-off. The topology in this case is: 2 PConv layers and filter size of 5×5. We compare to a closing with a square of size 2, empirically found to deliver best results. We make the task even harder by using larger-than-optimal support. Training is performed in fully on-line fashion. While the target images are kept fixed, the input is generated by adding random noise sample by sample. So the network never sees the same pattern twice. Figure 7–top compares the two approaches, and shows the noisy image. We see that learning substantially improves the PSNR measure.

We continue with an even more challenging task, a 10% salt'n'pepper denoising. The network is made of 4 PConv layers, a very long pipeline. We compare to an opening with a square of size 2×2 on a closing with the same structuring

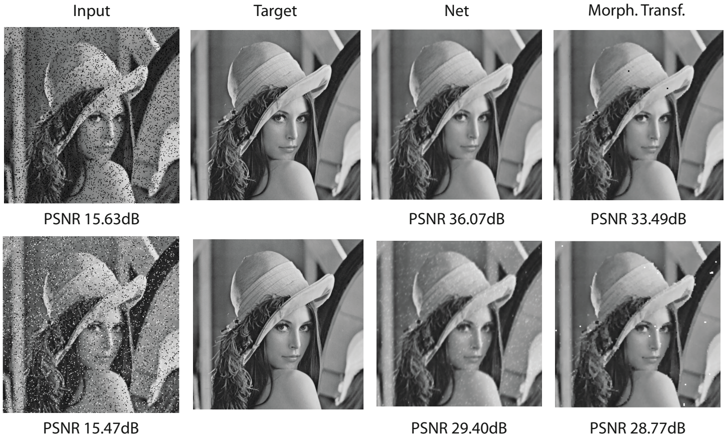


Fig. 7. *Top*: Binomial noise removal task. The learned nonlinear operator performs better than the hand-crafted one. Learning uses noisy and original images—there is no prior on the task. *Bottom*: Salt'n'pepper noise removal task. Even here, the learned operator performs better than the corresponding morphological one.

element. Training follows the same protocol as the one for the binomial noise. Images are generated online. This creates a possibly infinite dataset with very small memory footprint. Figure 7–bottom shows results. Although we can observe some limitations of our approach, it still exhibits the best PSNR also in this application.

Finally we consider the case of *total variation (TV)* restoration from an image corrupted by 6% additive Gaussian noise. Morphological filtering does not excel at this task. A MCNN is trained to learn the mapping from noisy image to TV restored image. How well can it approximate any target transformation with a pseudo-morphological pipeline? The architecture is composed of 2 PConv layers with 2 filters each plus an averaging layer. Results are shown in Figure 8.

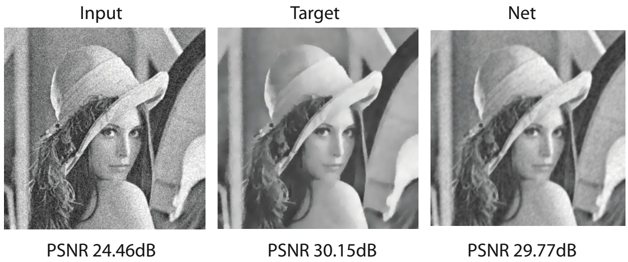


Fig. 8. Total Variation (TV) task. The network has to learn to approximate the TV output (target) by means of averaging two filtering pipelines.

5 Conclusion and Perspectives

Our MCNN for learning morphological operators is based on a novel PConv convolutional layer and inherits all the benefits of gradient-based deep-learning algorithms. It can easily learn complex topologies and operator chains such as white/black top-hats and we showed its application to steel defect detection. In future work we intend to let MCNN simultaneously learn banks of morphological filters and longer filtering pipelines.

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. Bullen, P.: Handbook of Means and Their Inequalities, 2nd edn. Springer (1987)
3. Cireşan, D.C., Meier, U., Masci, J., Schmidhuber, J.: A committee of neural networks for traffic sign classification. In: International Joint Conference on Neural Networks, IJCNN 2011 (2011)

4. Cireşan, D.C., Meier, U., Masci, J., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: International Joint Conference on Artificial Intelligence, IJCA I2011 (2011)
5. Ciresan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS (2012)
6. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Convolutional neural network committees for handwritten character classification. In: ICDAR, pp. 1250–1254 (2011)
7. Harvey, N.R., Marshall, S.: The use of genetic algorithms in morphological filter design. Signal Processing: Image Communication 8(1), 55–71 (1996)
8. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology 195(1), 215–243 (1968)
9. LeCun, Y.: Une procédure d'apprentissage pour réseau à seuil asymétrique. In: Proceedings of Cognitiva, Paris, vol. 85, pp. 599–604 (1985)
10. Masci, J., Meier, U., Cireşan, D.C., Fricout, G., Schmidhuber, J.: Steel defect classification with max-pooling convolutional neural networks. In: International Joint Conference on Neural Networks (2012)
11. Nakashizuka, M., Takenaka, S., Iiguni, Y.: Learning of structuring elements for morphological image model with a sparsity prior. In: IEEE International Conference on Image Processing, ICIP 2010, pp. 85–88 (2010)
12. Pessoa, L.F.C., Maragos, P.: Mrl-filters: a general class of nonlinear systems and their optimal design for image processing. IEEE Transactions on Image Processing 7(7), 966–978 (1998)
13. Salembier, P.: Adaptive rank order based filters. Signal Processing 27, 1–25 (1992)
14. Salembier, P.: Structuring element adaptation for morphological filters. J. of Visual Communication and Image Representation 3(2), 115–136 (1992)
15. Scherer, D., Müller, A., Behnke, S.: Evaluation of pooling operations in convolutional architectures for object recognition. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010, Part III. LNCS, vol. 6354, pp. 92–101. Springer, Heidelberg (2010)
16. Serra, J.: Image analysis and mathematical morphology. Academic Press, London (1982)
17. Soille, P.: Morphological Image Analysis: Principles and Applications. Springer (1999)
18. Turaga, S.C., Murray, J.F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., Denk, W., Seung, H.S.: Convolutional networks can learn to generate affinity graphs for image segmentation. Neural Comput. 22(2), 511–538 (2010)
19. van Vliet, L.J.: Robust local max-min filters by normalized power-weighted filtering. In: Proc. of IEEE ICPR 2004, vol. 1, pp. 696–699 (2004)
20. Werbos, P.J.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. thesis, Harvard University (1974)
21. Wilson, S.S.: Training structuring elements in morphological networks. In: Mathematical Morphology in Image Processing, ch. 1, pp. 1–42. Marcel Dekker (1993)

Flooding Edge or Node Weighted Graphs

Fernand Meyer

CMM-Centre de Morphologie Mathématique,
Mathématiques et Systèmes, MINES ParisTech, France
fernand.meyer@mines-paristech.fr

Abstract. This paper defines floodings on edge weighted and on node weighted graphs. Of particular interest are the highest floodings of a graph below a ceiling function defined on the nodes. It is shown that each flooding on a node weighted graph may be interpreted as a flooding on an edge weighted graphs with appropriate weights on the edges. The highest flooding of a graph under a ceiling function is then interpreted as a shortest distance on an augmented graph, using the ultrametric distance function. Thanks to this remark, the classical shortest distance algorithms may be used for constructing floodings.

1 Introduction

Reconstruction openings and closings are the simplest connected operators [2], [8], [6]. They are used for extracting marked particles in binary images or marked blobs in grey tone images. They are combined in morphological filters, in particular in levelings [9], [3]. Reconstruction closings act as floodings of a topographic surface and have all properties of a physical flooding. They are precious for simplifying gradient images, in filling unwanted catchment basins, on which a subsequent watershed transform extracts the targeted objects [4]. Flooding a topographic surface may thus be modeled as flooding a node weighted graph, with unweighted edges, the node weights representing the ground level. We call such a graph topographic graph or TG.

In the context of interactive segmentation, it often happens that watersheds have to be constructed: for each modification, addition of suppression of one or several markers, a new watershed has to be constructed. For this reason, it is advantageous to work at two scales. At the pixel level, a first watershed is constructed, with a catchment basin associated to each regional minimum. The information is then summarized in form of a region neighboring graph G, where each node represents a catchment basin and edges connect neighboring nodes. The edges are weighted by the altitude of the pass point between both adjacent regions. The objects to be segmented are selected as markers. The graph is flooded from sources placed at the marker positions and each node is assigned to the source by which it has been flooded.

So it appears that two types of floodings are often used in the context of segmentation or filtering, the one on node weighted graphs, the other on edge weighted graph. We analyze below the relations between both types. We then

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 341–352, 2013.

show that each flooding of a node weighted graph also is a flooding of an edge weighted graph with appropriate edge weights.

The highest flooding under a ceiling function may be interpreted as the shortest distance to the root for the ultrametric flooding distance in an augmented graph. The ultrametric distance between two nodes is the minimal altitude of a flooding for which both nodes are flooded. This remark permits to flood edge or node weighted graphs by using shortest path algorithms.

2 The Laws of Hydrostatics and Floodings

2.1 Edge Weighted Graphs

Consider a non oriented node and edge weighted graph $G = [E, N]$, E representing the edges and N the nodes.

In order to give a physical interpretation to our graph, we consider the nodes as vertical tanks of infinite height ; the tanks also have an infinite depth ; in other terms, there is no ground level. The weight τ_i represents the level of water in the tank i. Two neighboring tanks i and j are linked by a pipe at an altitude e_{ij} equal to the weight of the edge. We call such an edge weighted graph a tank network or TN.

We suppose that the laws of hydrostatics apply to our network of tanks and pipes:

- if the level τ_i in the tank i is higher than the pipe e_{ij}, then the levels is the same in both tanks i and j : $\tau_i = \tau_j$.
- the level τ_i in the tank i cannot be higher than the level τ_j, unless $e_{ij} \geq \tau_i$.

In fact this second condition implies the first one.

Definition 1. *The distribution τ of water in the pipes of the graph $[E, N]$ is a flooding of this graph, i.e. is a stable distribution of fluid if it verifies the*

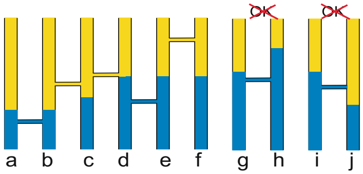


Fig. 1. Tank and pipe network:
- A and B form a regional minimum with $\tau_A = \tau_B = \lambda$; $e_{AB} \leq \lambda$; $e_{BC} > \lambda$
- B and C have unequal levels but are separated by a higher pipe.
- D and E form a full lake, reaching the level of its lowest exhaust pipe e_{CD}
- E and F have the same level ; however they do not form a lake, as they are linked by a pipe which is higher.
The distribution in the last four tanks is not compatible with the laws of hydrostatics.

criterion <for any couple of neighboring nodes (p, q) : $(\tau_p > \tau_q \Rightarrow e_{pq} \geq \tau_p)$ *(criterion* 1*)>*

Figure 1 presents a number of configurations compatible with the laws of hydrostatics and others which are not.

The following equivalence yield other useful criteria for recognizing flooding distributions on tank networks.

$(\tau_p > \tau_q \Rightarrow e_{pq} \geq \tau_p) \Leftrightarrow (\text{not } (\tau_p > \tau_q) \text{ or } e_{pq} \geq \tau_p) \Leftrightarrow$
$(\tau_p \leq \tau_q \text{ or } \tau_p \leq e_{pq}) \Leftrightarrow (\tau_p \leq \tau_q \vee e_{pq})$ *(criterion* 2*)*

2.2 Node Weighted Graphs

A topographic surface may be modeled by a node weighted graph. The node weights f_i indicate the ground level. The edges are not weighted: the existence of an edge (i, j) between the nodes i and j simply indicates that these nodes are neighbors. We call such a graph a topographic graph or TG.

A topographic graph is flooded if the nodes are assigned a second family of weights indicating the level of the flooding at each node. A distribution τ of node weights will represent an effective flooding if it verifies a number of conditions of equilibrium:

- A flooding being always above the ground level: $\tau_i \geq f_i$.
- As there is nothing to prevent the water to flow from a higher to a lower position, an inequal level of water at two neighboring nodes p and q is impossible, except when the highest node is dry ; hence $\tau_p > \tau_q \Rightarrow \tau_p = f_p$, indicating that the highest level is dry, without water. It is the case in fig. 2A but not in B, which is not physically possible.
- This criterion has the consequence that in a lake, the level of all nodes is the same. Hence, if p and q are two neighboring nodes verifying $\tau_p > f_p$ and $\tau_q > f_q$, then necessarily we have $\tau_p = \tau_q$.

We have shown ([4]) that if this criterion is verified, then τ is a flooding of f. For this reason, we take it as an axiom to be verified by a distribution τ to represent a flooding.

Definition 2. *The distribution τ of water on the nodes of the graph $[E, N]$ with unweighted edges and ground level f is a flooding of this graph, i.e. is a stable*

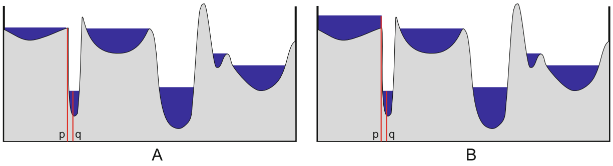


Fig. 2. A: a physically possible flooding ; B : an impossible flooding, where a lake is limited by a wall of water at position p

distribution of fluid if it verifies the criterion: $\tau \geq f$ *and for any couple of neighboring nodes* (p, q) *we have* $\tau_p > \tau_q \Rightarrow \tau_p = f_p$) (*criterion* 3)

2.3 Flooding a Topographic Graph = Flooding a Tank Network

Consider a topographic graph $G_n = [E, N]$ with a ground level distribution on the nodes equal to a function f, the edges being not weighted. If such a graph is flooded, the lowest level of flood covering two neighboring nodes p and q is equal to $f_p \vee f_q$. Consider now a second graph G_e with the same structure but with edge weights $e_{pq} = f_p \vee f_q$. These edge weights are the result of a dilation $e = \delta_{en} f$ assigning to each edge a weight equal to the maximal weight of its extremities.

Consider a flood distribution τ verifying $\tau \geq f$. We show that it is a valid flood distribution on the node weighted graph G_n if and only if it is a valid flood distribution on the edge weighted graph G_e with edge weights $e_{pq} = f_p \vee f_q$.

If it is a valid flooding on the edge weighted graph G_e, it verifies the criterion (1) : For any couple of neighboring nodes (p, q) we have: $(\tau_p > \tau_q \Rightarrow e_{pq} \geq \tau_p)$ which is equivalent with $\{\tau_p \leq \tau_q \text{ or } e_{pq} \geq \tau_p\} \Leftrightarrow \{\tau_p \leq \tau_q \vee e_{pq} = \tau_q \vee f_p \vee f_q = \tau_q \vee f_p\}$. as $\tau \geq f$.

If it is a valid flooding on the node weighted graph G_n, It verifies the criterion (3). As $\tau \geq f$, this criterion may be rephrased in $\{\tau_p > \tau_q \Rightarrow \tau_p \leq f_p\} \Leftrightarrow \{\tau_p \leq \tau_q$ or $\tau_p \leq f_p\} \Leftrightarrow \{\tau_p \leq \tau_q \vee f_p\}$.

Both criteria are identical, which shows that the flooding is a valid flooding for both graph G_n and G_e.

Theorem 1. *If* f *and* τ *are two weight distributions on the nodes of a graph and verify* $\tau \geq f$, *then* τ *is a valid flooding of graph considered as a topographic graph if and only if it is a valid flooding of the tank network with edge weights* $e_{pq} = f_p \vee f_q$.

This theorem has important consequences. A topographic graph with a ground level f immediately inherits all properties of tank networks, through the equivalent tank network with edge weights $e_{pq} = f_p \vee f_q$.

3 Lakes and Regional Minima

3.1 Lakes

A lake is a surface of water of uniform altitude, such that the addition of a drop of water increases the level of its total surface or produces an overflow. For this reason, in the case of tank networks, it is not sufficient that the nodes have a uniform altitude, but simultaneously, the inside edges should have an altitude which is lower or equal so as to permit the added drop of water to impact the whole surface. We first define an equivalence relation: pRq if and only if there exists a path $(n_1 = p, n_2, ...n_k = q)$ such that n_i and n_{i+1} are neighbors, all nodes on the path have the same flooding level λ and all edges (n_i, n_{i+1}) have weights $e_{ij} \leq \lambda$.

Tank networks: *A lake of a flooded tank network is a class of the equivalence relation R.*

Topographic graphs: *The preceding definition yields the simpler following definition for TG. A lake is a maximal connected component of nodes such that all couples (p, q) of neighboring pixels in the lake verify $\tau_p = \tau_q$.* A lake on a topographic graph is dry if it has a uniform altitude at the ground level. It is a wet lake, if it contains at least one pixel p for which $\tau_p > f_p$. If two neighboring nodes p and q verify $\tau_p > f_p$ and $\tau_q > f_q$, then $\tau_p = \tau_q$.

3.2 Regional Minima

Tank Networks. What happens at the boundary of a lake in a tank network? Consider 2 neighboring pixels (p, q), q being inside a lake of altitude λ and p outside. Being not both inside the lake, the relation pRq is not satisfied. They verify one of the following conditions: a) $\tau_p \neq \tau_q$ or b) $\tau_p = \tau_q = \lambda$ and along each path $(n_1 = p, n_2, ...n_k = q)$ of altitude λ there exists an edge (n_i, n_{i+1}) with a weight $e_{ij} > \lambda$. In particular $e_{pq} > \lambda$, as (p, q) is such a path. Thus in situation b) one has to climb for going out of the lake.

Consider now the condition a) $\tau_p \neq \tau_q$. There are two possibilities:
* $\tau_p > \tau_q = \lambda$ which implies $e_{pq} \geq \tau_p > \tau_q = \lambda$, one has to climb in order to cross the outgoing edge.
* $\tau_p < \tau_q = \lambda$ which implies $e_{pq} \geq \tau_q = \lambda$. If $e_{pq} > \tau_q$, then the outgoing edge is climbing. If $e_{pq} = \tau_q$, then the node q is an exhaust node for the lake towards a lower node p through the edge $e_{pq} = \lambda$.

Summarizing the situations a) and b): each outgoing edge (q, p) of a lake is either a climbing edge $e_{pq} > \lambda$ or an exhaust edge from q to p verifying $\tau_q = e_{pq} = \lambda > \tau_p$. In other terms, in a lake without exhaust edges, all outgoing edges are higher than the level of the lake. Such a lake is called regional minimum lake. A lake with one or several exhaust edges is called full lake. Adding a drop of water to a full lake provokes an overflow through the exhaust edges.

A regional minimum or a tank network is a lake with all outgoing edges, or cocycle edges having a higher altitude.

A lake of level λ on a tank network is a full lake, if there exists a couple of neighboring nodes p and q verifying $\tau_p < \tau_q = \lambda$ and $e_{pq} = \lambda$.

Topographic Graphs. Consider now a topographic graph G with ground level f and its derived tank network G' with edge weights $e_{pq} = f_p \vee f_q$. Let us apply the preceding definitions to the derived tank network G'.

A regional minimum of G' is a lake X with all outgoing edges, or cocycle edges having a higher altitude. Consider such an edge (q, p), q inside the lake at altitude λ and p outside the lake. Then $e_{pq} > \lambda$. But $e_{pq} = f_p \vee f_q$ and $\lambda = \tau_q \geq f_q$ imply $e_{pq} = f_p$. Inversely if all outside neighbors of a lake verify $f_p > \lambda$, then all edges of the cocycle of the lake verify $e_{pq} = f_p \vee f_q > \lambda$. This leads to the following classical definition: *A regional minimum is a lake for which the ground level of all outside neighbors has a higher altitude.*

A lake of the flooding of a topographic graph is a full lake, if there exists a couple of neighboring nodes p outside the lake and q inside $\tau_p < \tau_q = \lambda$ and $e_{pq} = f_p \vee f_q = \lambda$. But since $f_p \leq \tau_p < \lambda$, we have $\tau_q = f_q$. This leads to the characterization of a full lake: *A lake on a topographic surface is a full lake of altitude λ if there exist two neighboring nodes p inside the lake and q outside, such that $\tau_p < \tau_q = f_q$.*

A full lake touches the ground level at at least one node on its boundary, with a neighboring node where the flood is lower.

4 Constrained Highest Floodings on Node and Edge Weighted Graphs

There exist infinitely many distributions of valid floodings of the same graph G. If θ_1 and θ_2 are two floodings of the same graph (TN or TG), then $\theta_1 \vee \theta_2$ also is a flooding of this graph. For this reason, the supremum τ of all floodings or G below a ceiling function h also is flooding of G, this flooding is unique.

Any flooding θ verifies the relation: $\theta_p \leq \theta_q \vee e_{pq}$, for each neighbor q of p. As this relation is to be true for all neighbors of p, we have $\theta_p \leq \bigwedge_{q \text{ neighbor of } p} (\theta_q \vee e_{pq})$ Simultaneously $\theta_p \leq h_p$. So $\theta_p \leq h_p \wedge \bigwedge_{q \text{ neighbor of } p} (\theta_q \vee e_{pq})$ and the highest of them, τ verifies

$$\tau_p = h_p \wedge \bigwedge_{q \text{ neighbor of } p} (\tau_q \vee e_{pq}) \qquad (6).$$

Consider now a topographic graph G_n with a ground level f and a ceiling function $h \geq f$. The associated edge weighted graph or tank network has the edge weights $\delta_{en} f$. Any flooding $\theta \geq f$ valid for one of the graphs is valid for the other. In particular, the highest floodings below h for both graphs are identical. For this reason, for getting the highest flooding of the topographic graph G_n with node weights f under the ceiling function $h \geq f$, we simply may construct the highest flooding of the associated tank network G_e with the same edge weigths under the same ceiling function. The criterion 6, by replacing e_{pq} by its value, becomes $\tau_p = h_p \wedge \bigwedge_{q \text{ neighbor of } p} (\tau_q \vee f_p \vee f_q) = h_p \wedge \bigwedge_{q \text{ neighbor of } p} (\tau_q \vee f_p) =$

$$h_p \wedge \left[f_p \vee \bigwedge_{q \text{ neighbor of } p} (\tau_q) \right] \qquad (7)$$

The regional minima of constrained highest floodings

The level of the full lakes is solely determined by the altitude of the lowest pass point surrounding the lakes. In the regional minima lakes, the level cannot get higher, as it is blocked by the ceiling function h. More precisely : *Any regional minimum lake of the highest flooding of a tank network G_e with edge weights e, below a ceiling function h, contains a regional minimum lake of the flooding $[\delta_{en} h \vee e, h]$.*

In the case of a topographic graph G_n this theorem takes the simpler form: *Any regional minimum lake of the highest flooding of a graph G_n with node weights f, below a ceiling function h, contains a regional minimum lake of the function h.*

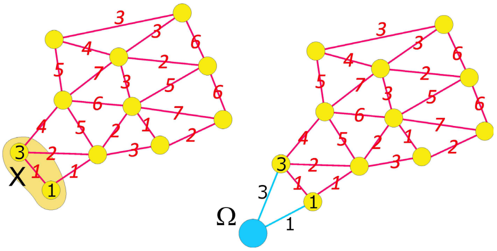


Fig. 3. Adding a dummy node linked to each node x in X by an edge weighted by the offset at x.

5 Algorithms for Computing the Highest Flooding on Node and Edge Weighted Graphs

5.1 Highest Floodings and Shortest Distances

We now add to the graph G_e a dummy node Ω with a weight $\tau_\Omega = 0$ linked by a dummy edge (Ω, p) with each node p and holding a weight equal to h_p. We get like that an augmented graph $\widehat{G_e}$. Relation (6) may be rewritten as $\tau_p = (\tau_\Omega \vee e_{\Omega p}) \wedge \bigwedge_{q \text{ neighbor of } p} (\tau_q \vee e_{qp})$. This formula expresses that the shortest path between Ω and p is $e_{\Omega p} = h_p$ if the path is simply the edge (Ω, p) or is equal to $(\tau_s \vee e_{ps})$ if the path passes through the neighbor s of p, and if $(\tau_q \vee e_{qp})$ takes its smallest value for $q = s$.

For the ultrametric flooding distance, the length of a path in an edge weighted graph is equal to the weight of the highest edge of the path. The shortest distance between two nodes p and q is the lowest highest edge of all paths between p and q in the algebra $(\min, \max)$ and may be constructed by any shortest algorithm published in the literature (see [1])

Theorem 2. *The highest flooding of the graph G below a function h defined on the nodes is the shortest distance of each node to Ω in the* $(\min, \max)$ *algebra.*

The Algorithm of Berge. Consider first a **tank network** G_e with edge weights $e_{pq.}$ This distribution being not a flooding, the algorithm recursively applies the relation (6), which lowers or keeps equal the node values, until stability is reached.

Initialisation: $\tau_p^{(0)} = h_p$

Repeat until $\tau_p^{(m)} = \tau_p^{(m-1)}$: $\tau_p^{(n)} = h_p \wedge \bigwedge_{q \text{ neighbor of } p} \left(\tau_q^{(n-1)} \vee e_{pq}\right)$

Stability is necessarily reached after a number n of iteration as the values of τ decrease and have a lower ceiling equal to 0. We then have $\tau_p^{(n)} = h_p \wedge \bigwedge_{q \text{ neighbor of } p} \left(\tau_q^{(n)} \vee e_{pq}\right)$, indicating that $\tau^{(n)}$ is a flooding of the graph G.

As $\tau_p^{(n)} \leq \tau_p^{(n-1)} \leq h_p$, we get an equivalent algorithm with the following sequence.
Repeat until $\tau_p^{(m)} = \tau_p^{(m-1)} : \tau_p^{(n)} = \tau_p^{(n-1)} \wedge \bigwedge_{q \text{ neighbor of } p} \left(\tau_q^{(n-1)} \vee e_{pq}\right)$.

Consider now **a topographic graph** G_n with a ground level f. The highest flooding of G_n under the node weights h also is the highest flooding of G_e with the same ceiling function h. For this reason the algorithm of Berge becomes, if one replaces e_{pq} by $f_p \vee f_q$ and remembers that $\tau \geq f$:

Initialisation: $\tau_p^{(0)} = h_p$

Repeat until $\tau_p^{(m)} = \tau_p^{(m-1)} : \tau_p^{(n)} = h_p \wedge \bigwedge_{q \text{ neighbor of } p} \left(\tau_q^{(n-1)} \vee f_p \vee f_q\right) = h_p \wedge \bigwedge_{q \text{ neighbor of } p} \left(\tau_q^{(n-1)} \vee f_p\right)$

or repeat until $\tau_p^{(m)} = \tau_p^{(m-1)} : \tau_p^{(n)} = \tau_p^{(n-1)} \wedge \bigwedge_{q \text{ neighbor of } p} \left(\tau_q^{(n-1)} \vee f_p\right)$

The Moore Dijkstra Shortest Path Algorithm [5]. This greedy algorithm takes as many steps as there are nodes. At any step, S represents the subset of nodes for which the shortest path is known. For any neighboring node of S, the length of the shortest path for which all edges but the last belong to S constitutes an overestimation of this length.

Initially, the only node whose flooding level is known is the dummy node Ω, with a level equal to 0. The best guess of the flooding level of all other nodes is $\tau_p = h_p$. For this reason the domain where the flooding level is guessed contains all nodes : $\overline{S} = N$. In fact, the dummy node is not necessary and we obtain the same result if we initialize by $S = \varnothing$.

Consider now a step of the ongoing the flooding. S contains all nodes whose flooding level is known, The estimation is made during the flooding, by assuming that the flooding arrives through one of its neighbors already in S. The node with the lowest guess has an estimated flooding level λ which is higher or equal than the flooding level of all nodes in S. And for this particular node, the guess is the correct flooding level. Indeed, if the flooding came through other neighbors, not yet in S, its value could not be lower : such a flood would have to quit S at a level higher or equal to λ and follow a path outside S along which it could become bigger. For this reason the node for which this estimation, say λ, is the lowest, has its flooding level correctly estimated. The algorithm proceeds by incorporating this node into S and updates its neighborhood.

Case of tank networks Initialization:

```
    $S = \varnothing$ ; $\overline{S} = N$ ; for each node $p$ in $N$ : $\tau_p = h_p$
Flooding:
While $\overline{S} \neq \varnothing$ repeat:
        Select $j \in \overline{S}$ for which $\tau_j = \min_{i \in \overline{S}} [\tau_i]$
        $\overline{S} = \overline{S} \backslash \{j\}$
        For any neighbor $i$ of $j$ in $\overline{S}$ do $\tau_i = \min [\tau_i, \tau_j \vee e_{ji}]$
End While
```

Progression of the flooding

The algorithm of Moore Dijkstra produces a flooding starting at level 0 at the node Ω.

The first node which is extracted from $\overline{S}$ is the node with the lowest estimation λ_1 representing the flooding level of this node. That means that the estimation of all nodes remaining in $\overline{S}$ is higher or equal than λ_1. Suppose that the $n-th$ node extracted is j with a flooding level λ_n, the smallest value of τ on $\overline{S}$. Each neighbor i of j which is updated with $\tau_i = \min[\tau_i, \tau_j \vee e_{ji}]$ remains above τ_j, since $\tau_i \geq \tau_j$ and $\tau_j \geq \tau_j \vee e_{ji}$. For this reason the level of flooding λ_n is never decreasing with n.

If we keep the edges linking each node with the node through which it has been flooded in the algorithm we get a tree. Along each edge of this tree, the level of the flood also is never decreasing.

Case of topographic graphs After replacing e_{pq} by its value $f_p \vee f_q$ and simplification, Dijkstra's algorithm become:

```
Initialization:
    $S = \varnothing$ ; for each node $p$ in $N$ : $\tau_p = h_p$
Flooding:
While $S \neq N$ repeat:
        Select $j \in \overline{S}$ for which $\tau_j = \min_{i \in \overline{S}} [\tau_i]$
        $S = S \cup \{j\}$
        For any neighbor $i$ of $j$ verifying $\tau_i > \tau_j \vee f_i$ do $\tau_i = \tau_j \vee f_i$
End While
```

Comments:

j is the smallest neighbor of i. If i is flooded through one of its neighbors, this neighbor can only be j and as as soon the value τ_i is computed once, this value is correct and final: $\tau_i = \tau_j \vee f_i$. Each node may be flagged as "guessed" at initialization and as "final" as soon its value has been updated in the core of the algorithm, as this value is now final.

5.2 Speeding Up the Algorithm of Dijkstra for Topographic Graphs

The core of the algorithm of Moore Dijkstra applies to any set S of nodes containing Ω and for which the flooding level is known. We define $\partial^+ S$ as the set of nodes q having a neighboring node in S. The flooding level of each node q in $\partial^+ S$ is estimated as follows : it is the smallest flooding value of q, if it is flooded through one of its neighbors in S. This value is $\widetilde{\tau}_q = \bigwedge_{\substack{p \in S \\ (p,q) neighbors}} \tau_p \vee e_{pq}$. If $p = \Omega$, then $\tau_p \vee e_{pq} = h_q$, otherwise $\tau_p \vee e_{pq} = \tau_p \vee f_q$

This estimation is the true flooding level for the node t for which the estimation is the lowest. Indeed, is the node t were flooded through a node q outside S, then its flooding path would quit S through another edge (u, v) of the cocycle of S, and the flooding level of t could only be higher.

This nice feature allows to expand S faster at the basic Dijkstra algorithm would do.

Consider a set S of nodes containing Ω and for which the flooding level is known. If for neighboring nodes p and q, we have $p \in S$, $p \neq \Omega$ and $q \notin S$, several configurations permit to determine directly the flooding level of q.

1. If $f_p = f_q$, then $\tau_p = \tau_q$, as the flooding is a connected operator.
2. If $f_p = h_p$, then ground and ceiling levels are identical, hence $\tau_p = f_p = h_p$
3. If $f_q \geq \tau_p$ then $\tau_q = f_q$. The proof is the following. For any neighboring nodes p and q the flooding levels verify $\tau_q \leq \tau_p \vee f_q = f_q$. But as $\tau_q \geq f_q$ we get $\tau_q = f_q$.
4. If $f_p \vee f_q < \tau_p$ then $\tau_q = f_q$. The proof is the following. $\{f_p \vee f_q < \tau_p\} \Leftrightarrow \{f_p < \tau_p \text{ and } f_q < \tau_p\}$. On the other hand the criterion for floodings $\{\tau_p > \tau_q\} \Rightarrow \{f_p = \tau_p\}$ is equivalent with $\{f_p < \tau_p\} \Rightarrow \{\tau_p \leq \tau_q\}$. And $\tau_q \leq \tau_p \vee f_q = \tau_p \vee f_p \vee f_q = \tau_p$ which shows that $\tau_p = \tau_q$
5. If $f_q < \tau_p$ then $\tau_q \leq f_q \vee \tau_p = \tau_{p.}$.
6. Suppose that (p, q) are neighbors and p is the node of $\partial^- S$ for which the flooding level τ_p is the lowest. If $f_q < \tau_p$, and if q is to be flooded by a node in S, this node necessarily is p and the estimate of the flooding level of q is $\tau_q = \tau_p \vee f_q = \tau_p$.

We may now derive a fast algorithm from these remarks. Suppose that during the flooding, the set S represents all flooded nodes and p is the node of $\partial^- S$ for which the flooding level τ_p is the lowest. Then, according rule 3, if $f_q \geq \tau_p$ then $\tau_q = f_q$. If on the contrary $f_q < \tau_p$, then the estimate of the flooding of q is $\tau_q = \tau_p$, which is the smallest estimate possible, except if there exists in $\overline{S}$ a node j with a smaller ceiling value h_j as τ_p. In this case, according to the algorithm of Dijkstra, the node h with the smaller estimation $h_j < \tau_p$ has to be introduced into S. This shows that in the case where $h_j \geq \tau_p$, all neighbors of p may be introduced at once into the set S, yielding the following algorithm:

A fast algorithm would then work as follows:

```
Initialization:
    $S = \varnothing$ ; for each node $p$ in $N$ : $\tau_p = h_p$
    for all other nodes $\tau_p = \infty$
Flooding:
While $S \neq N$ repeat:
        Select $j \in \overline{S}$ for which $\tau_j = \min_{i\in\overline{S}}[h_i]$ ; select $p \in \partial^- S$ for which $\tau_p = \min_{i\in\partial^- S}[\tau_i]$
        if $\tau_p > \tau_j$ then $S = S \cup \{j\}$ else
        For any neighbor $q$ of $p$ in $\overline{S}$ do $\tau_q = \tau_p \vee f_q$ and $S = S \cup \{q\}$
End While
```

Marker based segmentation. In the case of marker based segmentation each marker node gives birth to a region of the final segmentation. Each marker node p is dry and may be introduced in S at initialization. .

```
Initialization:
    $S = \varnothing$ ; for each marker node $p$ : $\tau_p = h_p$
Flooding:
```

```
While $S \neq N$ repeat:
        select $p \in \partial^- S$ for which $\tau_p = \min_{i \in \partial^- S} [\tau_i]$
        For any neighbor $q$ of $p$ in $\overline{S}$ do $\tau_q = \tau_p \vee f_q$ and $S = S \cup \{q\}$
End While
```

One obtains the classical segmentation algorithm with markers if one uses a hierarchical queue for scheduling the treatment.

Initialization:

Create a HQ and introduce each marker node p into it with a priority h_p and a specific label. This label will be assigned to the total region flooded through this marker

Flooding

```
    While the HQ is not empty repeat:
        (A) Extract from the HQ the node $j$ with the lowest prioriy
        Let $\lambda$ be this priority
            For any neighbor $i$ of $j$ without label
                introduce $i$ into the HQ with the priority $f_i \vee \lambda$ and the same label as $j$.
End While
```

This algorithm is exactly the same algorithm as the watershed algorithm associated to all regional minima of an image or a node weighted graph. With the same advantages, that is a selection of the geodesics with the highest steepness.

Flooding an Edge Weighted Graph Along Its Minimum Spanning Tree: The Algorithm of PRIM. Floodings on a tank network follow the laws of physics. If the nodes are partitioned into disjoint sets A and A^c, then a flood coming from A will pass through one of the lowest edges linking A and A^c. The algorithm of PRIM for constructing minimum spanning trees exploits this remark. It starts with an arbitrary node and constructs a tree by appending at each iteration the smallest edge of the cocycle to the tree. In the resulting tree, each node p is linked with Ω through a unique path. The flood coming from Ω necessarily follows this path.

Initialisation for a tank network

Initially, the tree T has only the node Ω and no edge: $T = \{\Omega, \varnothing\}$. $\tau_\Omega = 0$.

Expansion

```
As long the tree does not contain all nodes of the graph:
    Chose the lowest edge $(q, s)$ in the cocycle of $T$, such that $q \in T$ and $s \notin T$.
    Append the edge $(q, s)$ and the node $s$ to the tree: $T = T \cup \{s, (q, s)\}$
    $\tau_s = \tau_q \vee e_{qs}$
```

Case of a topographic graph The minimum spanning tree makes no sense for a node weighted graph G_n. It does if we consider G_e the associated edge weighted graph. By replacing e_{qs} by its value $f_q \vee f_s$, and since $\tau_q \geq f_q$, we get $\tau_q \vee f_q \vee f_s = \tau_q \vee f_s$. The flooding of the nodes and the construction of the tree may be done simultaneously.

6 Conclusion

The present study shows that there is no discrepancy between floodings on edge and node weighted graphs. Flooding a topographic graph with a ground level amounts at flooding a particular edge weighted graphs. Their highest floodings under a ceiling function are the same.

The same remark holds for the application of floodings to segmentation. It is equivalent to flood an image in order to fill all unwanted minima or to flood its region adjacency graph.

We did not concentrate on implementation issues and have presented the naked shortest distance algorithms. The scheduling of the operations may be taken in charge by hierarchical queues. We also highlighted several ways to dispatch the processing among different processors. For instance, if the flooding level at a node of a topographic graph is know, then all its neighbors with an altitude higher or equal than this node are dry. A processor may be assigned to the labeling of each dry zone. This process will be fast as it is independent of the value of the ceiling function which has not to be accessed anymore.

References

1. Gondran, M., Minoux, M.: Graphes et Algorithmes. Eyrolles (1995)
2. Klein, J.: Conception et Réalisation d'une unité logique pour l'analyse quantitative d'images. PhD thesis, University of Nancy (1976)
3. Meyer, F.: The levelings. In: Heijmans, H., Roerdink, J. (eds.) Mathematical Morphology and Its Applications to Image Processing, pp. 199–207. Kluwer (1998)
4. Meyer, F.: Flooding and segmentation. In: Goutsias, J., Vincent, L., Bloomberg, D.S. (eds.) Mathematical Morphology and its Applications to Image and Signal Processing. Computational Imaging and Vision, vol. 18, pp. 189–198 (2002)
5. Moore, E.F.: The shortest path through a maze. In: Proc. Int. Symposium on Theory of Switching, vol. 30, pp. 285–292 (1957)
6. Salembier, P., Serra, J.: Flat zones filtering, connected operators and filters by reconstruction. IEEE Transactions on Image Processing 3(8), 1153–1160 (1995)
7. Serra, J. (ed.): Image Analysis and Mathematical Morphology. II: Theoretical Advances. Academic Press, London (1988)
8. Vincent, L.: Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. IEEE Trans. in Image Procesing, 176–201 (1993)
9. Vachier, C., Vincent, L.: Valuation of image extrema using alternating filters by reconstruction. In: Proc. SPIE Image Algebra and Morphological Processing, San Diego CA (July 1995)

Towards Connected Filtering Based on Component-Graphs[⋆]

Benoît Naegel[1] and Nicolas Passat[2]

[1] ICube, UMR CNRS, Université de Strasbourg, France
[2] CReSTIC, EA 3804, Université de Reims, France

Abstract. In recent works, a new notion of component-graph has been introduced to extend the data structure of component-tree –and the induced antiextensive filtering methodologies– from grey-level images to multivalued ones. In this article, we briefly recall the main structural key-points of component-graphs, and we present the initial algorithmic results that open the way to the actual development of component-graph-based antiextensive filtering procedures.

Keywords: Component-graph, component-tree, multivalued images, partially ordered sets, connected operators, antiextensive filtering.

1 Introduction

Among the connected filtering approaches, that aim to process images without altering their contours, the component-tree [17] has received a specific attention. The component-tree is a data structure that models the characteristics of grey-level images by considering their successive binary level-sets. It is well-suited for processing grey-level images, based on hypotheses related to the topology (connectedness) and the specific intensity (local extrema) of structures of interest. In particular, it has been involved in several approaches (see, *e.g.*, [9,19]), especially for filtering and segmentation.

The success of the component-tree in the field of grey-level image processing, together with the increasing need for applications involving multivalued images, motivate its extension to the case of such images, which do not take their values in totally ordered sets, but in any (possibly partially) ordered ones.

After a preliminary study of the relations between component-trees and multivalued images [11], an extension of the component-tree to a more general notion of *component-graph* has been initiated in [13]. A study of the structural properties of these component-graphs has been proposed in [14], and we now consider the algorithmic key-points that will lead to the effective development of antiextensive filtering procedures.

This article is organised as follows. Sec. 2 briefly recalls some previous works on multivalued image handling in mathematical morphology. Sec. 3 describes the way to extend the notion of component-tree into a compliant notion of component-graph. The next sections explain how to build (Sec. 4), prune (Sec. 5) and recover a filtered multivalued image (Sec. 6) from a component-graph. The article is concluded by an illustrative example and perspective works, in Secs. 7 and 8, respectively.

[⋆] The research leading to these results has received funding from the French *Agence Nationale de la Recherche* (Grant Agreement ANR-2010-BLAN-0205 03).

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 353–364, 2013.

2 Previous Works

2.1 Mathematical Morphology and Multivalued Images

The extension of mathematical morphology –initially defined on binary, and then on grey-level images [8]– to multivalued (*e.g.*, colour, label, multi- and hyperspectral, etc.) images is an important task, motivated by potential applications in multiple areas. Several contributions have been devoted to this specific purpose. A whole state of the art is beyond the scope of this article, and the reader is referred to [3] for a recent survey.

By opposition to the grey-level case, the spaces in which such multivalued images take their values are not canonically equipped with total orders, but with partial ones. Several strategies have been considered to deal with this issue. Except in few works (see, *e.g.*, [15]), the proposed attempts generally consist of decomposing these value spaces into several totally ordered ones (marginal processing), or to define *ad hoc* total order relations on them (vectorial processing), with several variants (see, *e.g.*, [7,2,20]).

These approaches embed multivalued images into simpler frameworks, which authorise to process them similarly to grey-level ones, reducing in particular the complexity induced by partial orders. However, they also potentially bias the information intrinsically carried by these –more complex but richer– partially ordered value spaces.

2.2 The Case of Tree-Based Approaches

In the specific field of approaches based on tree structures (or more generally on partition hierarchies) the difficulties raised by multivalued images vary according to the proximity degree that exists between the data structure and the value space.

In the case of trees, or partition hierarchies, whose construction is not directly induced by the value space, and more precisely by its associated order (*e.g.*, partition trees [16], hierarchical watershed [6]), the use of intermediate functions (*e.g.*, a gradient for watershed, or more complex metrics for partition trees [18]) enables us to "hide" the complexity of the space, but necessarily induces a bias in the obtained data structure.

In the case of trees whose construction directly derives from the value space –and its order–, passing from total orders to partial ones leads to structural and algorithmic open issues. The main problems are caused by the fact that such data structures inherit from the structural complexity of the considered orders, and actually increase this complexity via their hierarchical structure. Among such kinds of trees, we find the component tree [17] and its autodual version, the tree of shapes [10]. Research about the extension to color images of the tree of shapes are currently developed by other authors [5].

In this article, we consider the component tree, and we investigate its extension to a more general hierarchical data structure that is no longer a tree, namely, the *component-graph*. In particular, we focus on the algorithmic consequences of this last property.

3 From Component-Trees to Component-Graphs

We now recall basic notions related to component-trees (Sec. 3.2). Then we introduce the recently proposed notion of component-graph [13,14] (Sec. 3.3). Due to space limitations, we present the minimal set of definitions and properties that are required to make this article self-contained. A more complete description may be found in [14].

3.1 Definitions and Hypotheses

Let Ω be a nonempty finite set equipped with a given connectivity. In particular, for any $X \subseteq \Omega$, the set of the connected components of X is noted $C[X]$.

Let V be a nonempty finite set equipped with an order relation $\leqslant$. We assume that $(V, \leqslant)$ admits a minimum, noted $\perp$.

Let I be an image defined on Ω and taking its values in V, *i.e.*, a function $I : \Omega \to V$. Without loss of generality, we can assume that $I^{-1}(\{\perp\}) = \{x \in \Omega \mid I(x) = \perp\} \neq \emptyset$.

For any $v \in V$, let $\lambda_v : V^{\Omega} \to 2^{\Omega}$ be the thresholding function at value v, defined for any image I, by $\lambda_v(I) = \{x \in \Omega \mid v \leqslant I(x)\}$.

3.2 Component-Trees

Here, we assume that $\leqslant$ is a total order. In other words, the image I is a grey-level image.

Let us define Ψ as the set of all the connected components obtained from all the thresholdings of I, that is

$$\Psi = \bigcup_{v \in V} C[\lambda_v(I)] \tag{1}$$

Definition 1 (Component-tree [17]) *The component-tree of I is the Hasse diagram $\mathfrak{T}$ of the partially ordered set* $(\Psi, \subseteq)$.

The component-tree has several virtues. Firstly, it can be built quite efficiently [17,12,4]. Secondly, it models the associated image in a lossless way. Indeed, we have

$$I = \bigvee_{v \in V}^{\leq} \; \bigvee_{X \in C[\lambda_v(I)]}^{\leq} C_{(X,v)} \tag{2}$$

where $\leq$ is the pointwise order relation on V^{Ω} induced by $\leqslant$, and $C_{(X,v)} : \Omega \to V$ is the cylinder function defined, for any $x \in \Omega$ by $C_{(X,v)}(x) = v$ if $x \in X$, and $\perp$ otherwise. Thirdly, any subset $\widehat{\Psi} \subseteq \Psi$ leads –by "substituting $\widehat{\Psi}$ to Ψ" in Eq. (2)– to a well-defined image $\widehat{I} : \Omega \to V$ that verifies $\widehat{I} \leq I$.

From these properties, an antiextensive filtering framework, based on component-trees, has been developed [17,9]. This framework, illustrated in Diagram (3), consists of three successive steps:

(i) the construction of the component-tree $\mathfrak{T}$ associated to I;
(ii) the pruning of $\mathfrak{T}$, based on an *ad hoc* criterion and a pruning policy, leading to a reduced component-tree $\widehat{\mathfrak{T}}$, corresponding to the Hasse diagram of $(\widehat{\Psi}, \subseteq)$; and
(iii) the reconstruction of the filtered image $\widehat{I} \leq I$ induced by $\widehat{\mathfrak{T}}$.

$$\begin{tikzcd}
I \arrow[r, "\text{Filtering}"] \arrow[d, "(i)"'] & \widehat{I} \leq I \\
\mathfrak{T} \arrow[r, "(ii)"] & \widehat{\mathfrak{T}} \arrow[u, "(iii)"']
\end{tikzcd} \tag{3}$$

The main purpose of this article is to provide algorithmic solutions (Secs. 4–6) for making this antiextensive filtering framework tractable in the case of component-graphs, that extend the component-trees to multivalued images. Before discussing such algorithmic issues, let us first introduce briefly this notion of component-graph.

3.3 Component-Graphs

We now relax the hypothesis of totality on $\leqslant$, which can then be either a total or a partial order. In Eq. (2), any cylinder function $C_{(X,v)}$ is generated by a couple (X, v) where $X \in \mathcal{C}[\lambda_v(I)]$ is a connected component of the thresholded image $\lambda_v(I) \subseteq \Omega$ of I at value v. In the sequel, (X, v) will be called a *valued connected component*. In particular, we define the set Θ of all the valued connected components of I as follows

$$\Theta = \bigcup_{v \in V} \mathcal{C}[\lambda_v(I)] \times \{v\} \tag{4}$$

From the order relation $\leqslant$ defined on V, and the inclusion relation $\subseteq$ on 2^{Ω}, we then define the order relation $\trianglelefteq$ on Θ as follows

$$(X_1, v_1) \trianglelefteq (X_2, v_2) \Longleftrightarrow (X_1 \subset X_2) \;\vee ((X_1 = X_2) \wedge (v_2 \leqslant v_1)) \tag{5}$$

In first approximation, the *component-graph* $\mathfrak{G}$ of I is the Hasse diagram of the ordered set $(\Theta, \trianglelefteq)$. However, three variants of component-graphs can relevantly be considered by defining two other subsets $\ddot{\Theta} \subseteq \dot{\Theta} \subseteq \Theta$ of valued connected components

$$\dot{\Theta} = \bigcup_{X \in \Psi} \{X\} \times \overset{\leqslant}{\bigtriangledown} \{v \mid X \in \mathcal{C}[\lambda_v(I)]\} \tag{6}$$

$$\ddot{\Theta} = \{(\Omega, \perp)\} \cup \bigcap \Big\{\Theta' \subseteq \Theta \;\Big|\; I = \overset{\leq}{\bigvee_{K \in \Theta'}} C_K\Big\} \tag{7}$$

where $\bigtriangledown$ denotes the set of the maximal elements. Broadly speaking, Θ gathers all the valued connected components induced by I; $\dot{\Theta}$ gathers the valued connected components of maximal values for any connected components; and $\ddot{\Theta}$ gathers the valued connected components associated to cylinders functions which are sup-generators of I. We note $\blacktriangleleft$ (resp. $\dot{\blacktriangleleft}$, resp. $\ddot{\blacktriangleleft}$) the cover relation associated to the order relation $\trianglelefteq$ on Θ (resp. to the restriction of $\trianglelefteq$ to $\dot{\Theta}$, resp. to the restriction of $\trianglelefteq$ to $\ddot{\Theta}$). We then have the following definition for the three variants of component-graphs.

Definition 2 (Component-graph(s) [14]) *The Θ- (resp. $\dot{\Theta}$-, resp. $\ddot{\Theta}$-)*component-graph *of I is the Hasse diagram $\mathfrak{G} = (\Theta, \blacktriangleleft)$ (resp. $\dot{\mathfrak{G}} = (\dot{\Theta}, \dot{\blacktriangleleft})$, resp. $\ddot{\mathfrak{G}} = (\ddot{\Theta}, \ddot{\blacktriangleleft})$) of the ordered set $(\Theta, \trianglelefteq)$ (resp. $(\dot{\Theta}, \trianglelefteq)$, resp. $(\ddot{\Theta}, \trianglelefteq)$). (The term $\mathring{\Theta}$-component-graph and the notation $\mathring{\mathfrak{G}} = (\mathring{\Theta}, \mathring{\blacktriangleleft})$ will sometimes be used for the three kinds of component-graphs.)*

The component-graph is a relevant extension of the component-tree, since (i) both notions are compliant for totally ordered sets $(V, \leqslant)$, and (ii) the component-graph satisfies the image (de)composition model associated to component-tree, defined in Eq. (2).

Property 1 ([14]) *If $\leqslant$ is a total order, then two of the three variants of component-graphs, namely $\dot{\mathfrak{G}}$ and $\ddot{\mathfrak{G}}$, are isomorphic to the component-tree $\mathfrak{T}$.*

Property 2 ([14]) *For the three variants of component-graphs, we have*

$$I = \overset{\leq}{\bigvee_{v \in V}} \; \overset{\leq}{\bigvee_{X \in \mathcal{C}[\lambda_v(I)]}} C_{(X,v)} = \overset{\leq}{\bigvee_{K \in \mathring{\Theta}}} C_K \tag{8}$$

4 Building the ($\ddot{\Theta}$-)Component-Graph

Efficient algorithms [17,12,4] have been proposed to build the component-tree, leading to algorithmic complexities which are nearly linear with respect to the image size. Such a linear bound is hard to reach in the case of multivalued images, in particular due to the structural properties of $\leqslant$, whose Hasse diagram is not necessarily a chain.

In the sequel, we specifically deal with the construction of the $\ddot{\Theta}$-component-graph. Our motivation is twofold. Firstly, the $\ddot{\Theta}$-component-graph is the only of the three variants that guarantees to avoid the appearance of new values in the filtered images, since any valued connected component of $\ddot{\Theta}$ actually contributes to the formation of the image I (see Eq. (7)). Secondly, due to the increasing cardinality of $\ddot{\Theta}$-, $\dot{\Theta}$- and Θ, the algorithmic process that is considered for building the $\ddot{\Theta}$-component-graph may be further used as a basis to develop (more complex) algorithms for building the other variants.

4.1 Algorithmics

Algorithm 1 describes the main procedure to compute the $\ddot{\Theta}$-component-graph. The collection of valued connected components (the *nodes* of the graph) is maintained using Tarjan's union-find algorithm, based on the `makeSet`, `find` and `link` operations, similarly to [12]. The array `graph` stores, for each canonical element p , the *set* of fathers of the node of p. To avoid to insert the same link twice, each cell of `graph` is managed as a *set* data structure. The pixels are processed by decreasing values, using a priority queue `pq`. More precisely (since the values are not totally ordered) a pixel of value v is processed only if all the pixels of values $v' > v$ have been processed.

A key point of the algorithm relies on the `lowestNodes` function. This function returns the set of minimal ancestors (the *lowest* nodes which are not comparable) of a node given a value. This function plays the same role as the array `lowestNode` in the component-tree computation, in Najman and Couprie's algorithm [12]. However, in the case of the component-tree, the `lowestNode` array can be maintained efficiently: by contrast, in the case of the component-graph, the `lowestNodes` function must be recomputed each time since its result depends on the value given in parameters.

4.2 Example

We illustrate the steps of the algorithm on a toy example. Fig. 1(a) depicts the Hasse diagram of the partially ordered set $(V, \leqslant)$ used by the image I (Fig. 1(b)). Successive threshold sets of I are depicted on Fig. 1(c–g).

Let us suppose that we have processed the pixels[1] having values f, d and b. The current computed graph is depicted on Fig. 2(a). The level c is processed, and the pixel 4 is extracted from the priority queue. It has four processed neighbors: $\{2, 3, 5, 6\}$. We observe that 4 and 2 are not comparable, while $I(4) < I(3)$; then 4 is an ancestor of 3. The result of the function `lowestNodes(3,`c`)` is the set $\{3\}$, so the pixel 4 is a direct father of 3. It is inserted in the set `graph[3]`. Similarly, the pixel 4 is a direct father of

[1] There is no pixel having the value e; then this value is not present in the $\ddot{\Theta}$-component-graph.

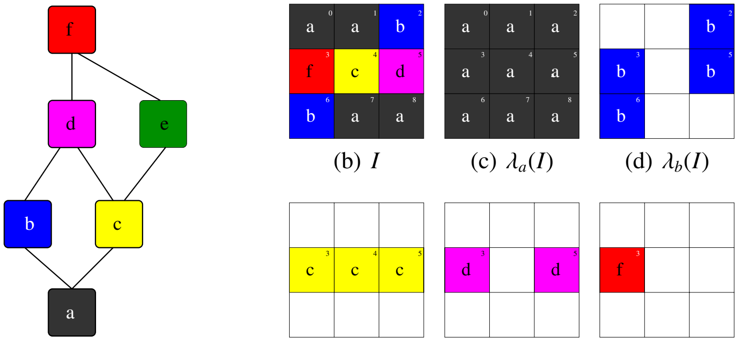
Algorithm 1. Computation of the $\ddot{\Theta}$-component-graph

```
Data: image $I : \Omega \to V$
Result: $\ddot{\Theta}$-component-graph $\ddot{\mathfrak{G}}$ (array $[0...N-1]$ storing, for each canonical element, the
        set of its “fathers”)
foreach $p \in \Omega$ do
    makeSet($p$);
    pq.put($p$,$I(p)$);
while pq $\neq \emptyset$ do
    $p \leftarrow$ pq.front();
    foreach already processed neighbors $q$ of $p$ do
        $adjNode \leftarrow$find($q$);
        if $I(adjNode) = I(p) \wedge adjNode \neq p$ then
            link($adjNode$,$p$); // $p$ is the new canonical element of the
                node
        else if $I(adjNode) < I(p)$ then
            nodesList $\leftarrow$ lowestNodes($adjNode$,$I(p)$);
            foreach $n \in$ nodesList do
                if $I(p) < I(n)$ then graph[$n$].insert($p$);
                else link($n$,$p$); // $p$ is the new canonical element of the
                    node
```

Function lowestNodes(node,value)

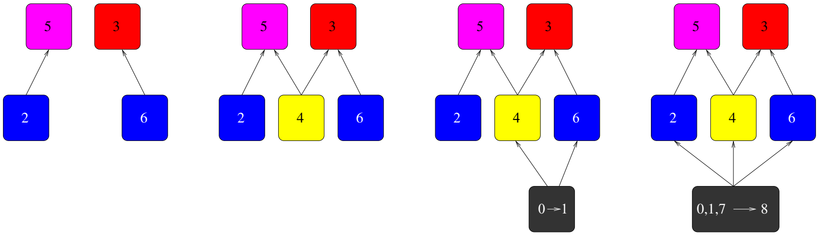
```
Data: active (array $[0...N-1]$ initialized to true)
Data: lowestNode (array $[0...N-1]$ initialized to true)
Result: nodesList: lowest ancestors of node having a value superior or equal to value
fifo.push(node);
while fifo $\neq \emptyset$ do
    $p \leftarrow$fifo.get();
    foreach father $f \in$ graph[$p$] do
        $q \leftarrow$ find($f$);
        if $I(q) \geqslant v \wedge$ active[$q$] then
            fifo.push($q$);
            active[$q$]$\leftarrow$ false;
        if active[$q$]= false then lowestNode[$p$]= false;
    if lowestNode[$p$]= true then nodesList.add($p$);
return nodesList
```

5 (see Fig. 2(b)). The level a is now processed. The point 0 is extracted. It has two processed neighbors of higher value: $\{3, 4\}$. The result of the function lowestNodes(3,a) is the set $\{4, 6\}$: 0 is then a direct father of these points (which are canonical elements). The result of the function lowestNodes(4,a) is the set $\{0\}$ (since 0 is now a father of 4 and 6), *i.e.*, the current visited point: nothing has to be done. The point 1 is processed and compared to its five neighbors $\{0, 2, 3, 4, 5\}$. We have $I(1) = I(0)$, then it



(a) $(V, \leqslant)$ (b) I (c) $\lambda_a(I)$ (d) $\lambda_b(I)$ (e) $\lambda_c(I)$ (f) $\lambda_d(I)$ (g) $\lambda_e(I)$

Fig. 1. (a) Hasse diagram of $(V, \leqslant)$. (b) Image I. (c–g) Components of $\ddot{\Theta}$-component-graph.



(a) Proc. level b (b) Proc. level c (c) Proc. level a (d) $\ddot{\mathfrak{G}}$

Fig. 2. $\ddot{\Theta}$-component-graph computation: illustration of some steps

becomes the canonical element of the (partial) node $\{0, 1\}$. We have $I(1) < I(2)$ and `lowestNodes(2,`a`)`$= \{2\}$, consequently 1 becomes a father of 2. The other comparisons do not change the graph, since `lowestNodes(`i`,`a`)`$= \{1\}$ for $i = 3, 4, 5$ (see Fig. 2(c)). Finally, the points 7 and 8 are processed. When 7 is compared to any of its neighbors of greater value, the function `lowestNodes` returns the set $\{1\}$. Since 7 and 1 have the same value but belong to different nodes, 7 and 1 are linked to the same node. At last, 8 is linked to 7 and becomes the new canonical element of the node $\{0, 1, 7, 8\}$. The final $(\ddot{\Theta})$-component-graph is depicted in Fig. 2(d).

5 Pruning the Component-Graph

Similarly to the antiextensive filtering framework based on component-trees, filtering with component-graphs requires to define a subset $\widehat{\Theta} \subseteq \mathring{\Theta}$. This choice is based on (i) a selection (Boolean) criterion ρ on $\mathring{\Theta}$, and (ii) a pruning policy which determines, together with ρ, which parts of the component-graph should be preserved.

Pruning Policies. If ρ is a non-increasing criterion, then various pruning policies can be considered. For component-trees, several classical policies have been defined (for example min, direct, max, subtractive, Viterbi [17,21]). In the case of component-graphs, the direct and max policies can be directly transposed, while the min one leads to two variants, $\min_1$ and $\min_2$, that can be axiomatically (and recursively) defined by

$$\rho(K_1) \Longrightarrow ((\forall K_2 \blacktriangleright K_1, K_2 \in \widehat{\Theta}_{\mathrm{min}_1}) \Rightarrow K_1 \in \widehat{\Theta}_{\mathrm{min}_1}) \tag{9}$$

$$\rho(K_1) \Longrightarrow ((\exists K_2 \blacktriangleright K_1, K_2 \in \widehat{\Theta}_{\mathrm{min}_2}) \Rightarrow K_1 \in \widehat{\Theta}_{\mathrm{min}_2}) \tag{10}$$

These four policies lead to increasing results, *i.e.*, for a same criterion ρ, we have

$$\widehat{\Theta}_{\mathrm{min}_1} \subseteq \widehat{\Theta}_{\mathrm{min}_2} \subseteq \widehat{\Theta}_{\mathrm{direct}} \subseteq \widehat{\Theta}_{\mathrm{max}} \tag{11}$$

Moreover, in the case where $(\mathring{\Theta}, \mathring{\blacktriangleleft})$ has a tree structure, the $\min_1$ and $\min_2$ policies are equivalent. In this case, which happens in particular when $(V, \leqslant)$ is totally ordered (*i.e.*, for grey-level images), we retrieve the standard min policy defined for component-trees.

Algorithmic Remarks. From their very definition, the $\min_1$ and $\min_2$ policies require to define $\widehat{\Theta}$ in a top-down fashion, *i.e.*, by starting from $(\Omega, \perp)$. By contrast, the max policy requires a bottom-up strategy. The direct policy –which can be applied indifferently in both directions– may be more relevantly involved in a bottom-up strategy.

The min₁ and max Pruning. The pruning of a component-graph based on the $\min_1$ and max policies is globally straightforward, since the edges of the pruned component-graph constitute a subset of the edges of the initial component-graph. This property leads in particular to pruning procedures whose algorithmic complexity is $\mathcal{O}(|\mathring{\Theta}| + |\mathring{\blacktriangleleft}|)$.

The min₂ and Direct Pruning. In the case where $(\mathring{\Theta}, \mathring{\blacktriangleleft})$ is not a tree, by opposition to the previous two policies, the $\min_2$ and direct ones do not imply that $\widehat{\blacktriangleleft} \subseteq \mathring{\blacktriangleleft}$. Indeed, when a element $K \in \mathring{\Theta}$ is not preserved in $\widehat{\Theta}$, each pair of edges of the form $K' \mathring{\blacktriangleleft} K \mathring{\blacktriangleleft} K''$ may lead to an edge $K' \mathrel{\widehat{\blacktriangleleft}} K''$. However, the existence of such an edge in $(\widehat{\Theta}, \widehat{\blacktriangleleft})$ is conditioned by the non-existence of a series of edges $K' \mathrel{\widehat{\blacktriangleleft}} K_1 \mathrel{\widehat{\blacktriangleleft}} \ldots \mathrel{\widehat{\blacktriangleleft}} K_t \mathrel{\widehat{\blacktriangleleft}} K''$. Based on these considerations, a relevant approach consists of first computing $(\widehat{\Theta}, \overline{\blacktriangleleft})$, where $\overline{\blacktriangleleft}$ is a superset of $\widehat{\blacktriangleleft}$, containing redundant edges which may be obtained by transitivity from the edges of $\widehat{\blacktriangleleft}$. Such an approach, that presents an algorithmic complexity $\mathcal{O}(|\trianglelefteq|)$, may then be followed by a standard transitive reduction procedure [1] to recover $\widehat{\blacktriangleleft}$ from $\overline{\blacktriangleleft}$.

6 Recovering a Filtered Image from a Pruned Component-Graph

Once $\widehat{\Theta}$ is defined, the filtered image $\widehat{I} : \Omega \to V$ should be obtained from the cylinder functions $\{C_K \mid K \in \widehat{\Theta}\}$. However, the expression of $\widehat{I}$ via Eq. (8) (by substituting $\widehat{I}$ to I, and $\widehat{\Theta}$ to $\mathring{\Theta}$) is not necessarily well-defined. Indeed, there is no guarantee that for any $x \in \Omega$, the set $\{C_K(x) \mid K \in \widehat{\Theta}\} \subseteq V$ admits a maximum (or even a supremum) for $\leqslant$.

In such conditions, it is then necessary to define a reconstructed image $\widetilde{I} : \Omega \to V$ formed by a set $\widetilde{\Theta}$ being "as similar as possible" to $\widehat{\Theta}$. In this first study, we chose to consider the sets $\widetilde{\Theta}$ leading to reconstructed images $\widetilde{I}$ being either greater or lower than the putative image $\widehat{I}$ (w.r.t. $\leq$). To this end, let us first define the following notions.

6.1 Well-Defined Sets of Valued Connected Components

We say that $\Theta' \subseteq \mathring{\Theta}$ is *well-defined* if $\bigvee^{\leqslant}_{K\in\Theta'} C_K$ exists, *i.e.*, if for any $x \in \Omega$, the set $\{C_K(x) \mid K \in \Theta'\}$ admits a maximum for $\leqslant$. We note $\mathring{\Xi} \subseteq 2^{\mathring{\Theta}}$ the set of all the well-defined subsets of $\mathring{\Theta}$, and for any $\Theta' \in \mathring{\Xi}$ we note $I_{\Theta'} = \bigvee^{\leqslant}_{K\in\Theta'} C_K$, namely the image reconstructed from Θ'. Let $\sim$ be the equivalence relation on $\mathring{\Xi}$ defined by

$$(\Theta' \sim \Theta'') \Longleftrightarrow (I_{\Theta'} = I_{\Theta''}) \tag{12}$$

that gathers the well-defined sets of valued connected components which lead to similar images. Let $\sqsubseteq$ by the (partial) relation order on the quotient set $\mathring{\Xi}/\!\sim$, defined by

$$([\Theta']_\sim \sqsubseteq [\Theta'']_\sim) \Longleftrightarrow (I_{\Theta'} \leq I_{\Theta''}) \tag{13}$$

that embeds the relation $\leq$ on images into the space of the (sets of) generating valued connected components. Given a subset of valued connected components $\widehat{\Theta} \subseteq \mathring{\Theta}$, we set

$$\mathring{\Xi}^{+}(\widehat{\Theta}) = \{\Theta' \in \mathring{\Xi} \mid \forall K \in \widehat{\Theta}, C_K \leq I_{\Theta'}\} \tag{14}$$

$$\mathring{\Xi}^{-}(\widehat{\Theta}) = \{\Theta' \in \mathring{\Xi} \mid \forall K \in \widehat{\Theta}, I_{\Theta'} \leq C_K\} \tag{15}$$

Since our purpose is to define a result image $\widetilde{I}$ being "as close as possible" to the putative image $\widehat{I}$, the choice of the solution set of valued connected components has to be made among the minimal (resp. maximal) equivalence classes $[\Theta']_\sim$, associated to the sets Θ' of $\mathring{\Xi}^{+}(\widehat{\Theta})$ (resp. $\mathring{\Xi}^{-}(\widehat{\Theta})$), for $\sqsubseteq$. More precisely, in order to respect both the content of the image I, and the nature of the component-graph ($\mathfrak{G}$, $\dot{\mathfrak{G}}$, or $\ddot{\mathfrak{G}}$), the sets Θ' have actually to be considered within $\mathring{\Xi}^{-}(\widehat{\Theta}) \cap 2^{\widehat{\Theta}}$ (resp. $\mathring{\Xi}^{+}(\widehat{\Theta}) \cap 2^{\mathring{\Theta}}$).

Note that in the case where a solution $\widetilde{\Theta}^{+}$ is determined among $\mathring{\Xi}^{+}(\widehat{\Theta})$ (*i.e.*, when $\widetilde{I}$ is greater than the putative image $\widehat{I}$), we do not necessarily have $\widehat{\Theta} \subseteq \widetilde{\Theta}^{+}$. However, we do have $(\widehat{\Theta} \cup \widetilde{\Theta}^{+}) \in \mathring{\Xi}^{+}(\widehat{\Theta})$, and $(\widehat{\Theta} \cup \widetilde{\Theta}^{+}) \sim \widetilde{\Theta}^{+}$. Broadly speaking (an equivalent version of) $\widetilde{\Theta}^{+}$ can be defined from $\widehat{\Theta}$ by relevantly adding new valued connected components.

Under such assumptions, we have in particular

$$[\emptyset]_\sim = [\{(\Omega, \bot)\}]_\sim \sqsubseteq [\widetilde{\Theta}^{-}]_\sim \sqsubseteq [\widehat{\Theta}]_\sim \sqsubseteq [\widehat{\Theta} \cup \widetilde{\Theta}^{+}]_\sim = [\widetilde{\Theta}^{+}]_\sim \sqsubseteq [\mathring{\Theta}]_\sim \tag{16}$$

The pruned set $\widehat{\Theta}$, obtained from $\mathring{\Theta}$, leads to a partition of Ω into two sets: $\Omega_w(\widehat{\Theta})$ that contains the points x such that $\{C_K(x) \mid K \in \widehat{\Theta}\}$ admits a maximum for $\leqslant$, and $\Omega_{\overline{w}}(\widehat{\Theta})$ that contains the points x such that $\{C_K(x) \mid K \in \widehat{\Theta}\}$ has several maximal elements for $\leqslant$, *i.e.*, where $\widehat{I}$ is not well-defined. We now discuss the way to deal with $\Omega_{\overline{w}}(\widehat{\Theta})$.

6.2 Algorithmics

Let us first assume that $\leqslant$ is a lower piecewise total order (LPTO), *i.e.*, that $(\{v' \in V \mid v' \leqslant v\}, \leqslant)$ is totally ordered for any $v \in V$. In such conditions, a component-graph has a tree structure [14], and we have the following property.

Property 3 *If $\leqslant$ is a LPTO, we have $\mathring{\Xi} = 2^{\mathring{\Theta}}$ and $[\widehat{\Theta}]_\sim = \bigvee^{\sqsubseteq}[\mathring{\Xi}^{-}(\widehat{\Theta})]_\sim = \bigwedge^{\sqsubseteq}[\mathring{\Xi}^{+}(\widehat{\Theta})]_\sim$.*

In other words, $\Omega_{\overline{w}}(\widehat{\Theta}) = \emptyset$, and $\widehat{I} = \widetilde{I}$ is then straightforwardly obtained from $\widehat{\Theta}$ by applying Eq. (8). In the sequel, we now suppose that $\leqslant$ is no longer a LPTO.

Reconstruction from $\boldsymbol{\mathring{\Xi}^{+}(\widehat{\Theta})}$. Let us first assume that $\leqslant$ is a lower piecewise lattice (LPL), *i.e.*, that $(\{v' \in V \mid v' \leqslant v\}, \leqslant)$ is a lattice for any $v \in V$. In such conditions, we have the following property for the Θ and the $\dot{\Theta}$-component-graphs.

Property 4 *If $\leqslant$ is a LPL, then $\bigwedge^{\sqsubseteq}[\Xi^{+}(\widehat{\Theta})]_{\sim}$ and $\bigwedge^{\sqsubseteq}[\dot{\Xi}^{+}(\widehat{\Theta})]_{\sim}$ exist. Moreover, in the case of the Θ-component-graph, $\widetilde{I} = I_{\widetilde{\Theta}^{+}}$ is directly defined from $\widehat{\Theta}$, by applying Eq. (8).*

In other words, $\Omega_{\overline{w}}(\widehat{\Theta}) \neq \emptyset$ in general, but the (unique) solution $\widetilde{I}$ can be straightforwardly obtained from $\widetilde{\Theta}^{+}$ in the case of $\mathfrak{G}$, and from $\widetilde{\Theta}^{+}$ and $\dot{\mathfrak{G}}$ in the case of $\dot{\mathfrak{G}}$.

For $\ddot{\mathfrak{G}}$ (and for any $\mathring{\mathfrak{G}}$ when $\leqslant$ is not a LPL), the definition of a solution $\widetilde{I}$ is no longer straightforward. Such a solution $\widetilde{I}$ can be obtained by defining $\widetilde{\Theta}^{+}$ by the following (non-deterministic) process. (For the sake of concision in the next property, we note the set of valued connected components of Θ' that are in conflict at a given point $x \in \Omega_{\overline{w}}(\Theta')$ as $\Theta'_x = \{(K', v') \in \Theta' \mid (x \in K') \wedge (v' \in \bigvee^{\leqslant}\{v'' \mid ((K'', v'') \in \Theta') \wedge (x \in K'')\})\}$.)

Property 5 *A set $\widetilde{\Theta}^{+}$ can be defined as $\mathcal{F}^{+}(\widehat{\Theta})$ where $\mathcal{F}^{+} : 2^{\mathring{\Theta}} \to 2^{\mathring{\Theta}}$ is the extensive function (recursively) defined by*

$$\mathcal{F}^{+}(\Theta') = \begin{cases} \Theta' & \text{if } \Omega_{\overline{w}}(\Theta') = \emptyset \\ \mathcal{F}(\Theta' \cup \{(K, v)\}) & \text{if } \Omega_{\overline{w}}(\Theta') \neq \emptyset \end{cases} \tag{17}$$

where $x \in \arg_y \min |\Theta'_y|$, and $(K, v) \in \bigwedge^{\trianglelefteq}\{(K', v') \mid (\forall (K'', v''), (K'', v'') \trianglelefteq (K', v')) \wedge (x \in K')\}$

It has to be noted that in the case of $\ddot{\mathfrak{G}}$, this process is deterministic, and the solution $\widetilde{I}$ is then unique. However, this is not the case for $\mathfrak{G}$ and $\dot{\mathfrak{G}}$.

Reconstruction from $\boldsymbol{\mathring{\Xi}^{-}(\widehat{\Theta})}$. When $\widetilde{\Theta}^{-}$ is determined among $\mathring{\Xi}^{-}(\widehat{\Theta})$, the solution $\widetilde{I}$ is not unique in general, independently from hypotheses about the kind of component-graph ($\mathfrak{G}$, $\dot{\mathfrak{G}}$, $\ddot{\mathfrak{G}}$), and the kind of order $\leqslant$ (except in the case of LPTO). Indeed, a solution $\widetilde{I}$ can be obtained by defining $\widetilde{\Theta}^{-}$ by the following (non-deterministic) process.

Property 6 *A set $\widetilde{\Theta}^{-}$ can be defined as $\mathcal{F}^{-}(\widehat{\Theta})$ where $\mathcal{F}^{-} : 2^{\widehat{\Theta}} \to 2^{\widehat{\Theta}}$ is the antiextensive function (recursively) defined by*

$$\mathcal{F}^{-}(\Theta') = \begin{cases} \Theta' & \text{if } \Omega_{\overline{w}}(\Theta') = \emptyset \\ \mathcal{F}^{-}(\Theta' \setminus \mathcal{X}) & \text{if } \Omega_{\overline{w}}(\Theta') \neq \emptyset \end{cases} \tag{18}$$

where $\mathcal{X} = \{(K', v') \in \Theta' \mid (K' \cap K \cap \Omega_{\overline{w}}(\Theta') \neq \emptyset) \wedge (v' \nleqslant v)\}$ and $(K, v) \in \Theta'$ verifies $K \cap \Omega_{\overline{w}}(\Theta') \neq \emptyset$ and $v \in \bigvee^{\leqslant}\{v' \mid ((K', v') \in \Theta') \wedge (K' \cap \Omega_{\overline{w}}(\Theta') \neq \emptyset)\}$.

7 An Application Example

In this section, we illustrate, on a simple application case, the interest of component-graphs for multimodal imaging. The proposed example involves both PET (Positron Emission Tomography) and standard CT (Computed Tomography) X-ray data. The CT

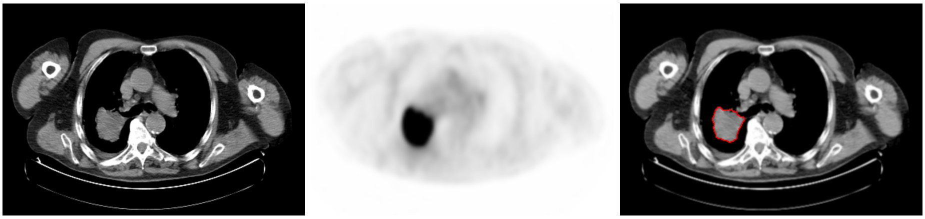


Fig. 3. (a) CT image. (b) PET image. (c) Detected components. (Image courtesy of D. Papathanassiou, Institut Jean-Godinot, France.)

image provides homogeneous zones that characterise specific tissues and organs. The PET image provides local intensity minima where tumours are active, but with a spatial accuracy that is lower that CT information. Consequently, by coupling both grey-level value spaces into a single value space V, it may be possible to extract some valued connected components that gather the spatial accuracy of CT images and the spectral accuracy of PET ones, thus leading to accurate localisation of tumors.

In the considered example (Fig 3(a,b)), the resolution of the image is 1318×864 and $V = [0, 255] \times [0, 255]$. Since the purpose is to extract bright objects in one image and dark objects in the other one, we consider the partial order relation $\leqslant$ defined by: $(v_1, v_2) \leqslant (w_1, w_2) \Leftrightarrow (v_1 \leq w_1) \wedge (v_2 \geq w_2)$. The graph $\ddot{\mathfrak{G}}$ is computed[2] and we consider for each node the attributes "area" and "height"[3]. Graph pruning is performed by using a non-increasing criterion based on (minimal and maximal) thresholds on the attributes. The reconstruction is performed "from bottom" based on the set $\ddot{\Xi}^{-}(\widehat{\Theta})$, therefore ensuring the removal of all non-desired components. Fig. 3(c) shows the detected areas.

One may notice that using the same strategy on the component-tree of the CT-scan image is not sufficient to extract the component, while using only the PET image prevents the extraction of an accurate contour.

8 Conclusion

This article has proposed first algorithmic results that may lead to connected filtering methodologies relying on component-graphs, and thus handling multivalued images (and, more generally, any valued graph structures [22]) in a "component-tree" fashion. Some issues remain however to be considered on the way toward such methodologies.

From an algorithmic point of view, solutions to (smartly) define both Θ- and $\dot{\Theta}$-component-graphs still have to be found. Distributed strategies [21] may provide some solutions to deal with these issues. Moreover, the reduction of the algorithmic complexity of filtered image reconstruction(s), in the most general cases, also has to be carefully considered. To this end, discrete optimisation strategies may be investigated.

[2] The graph computation takes 50 s on an Intel Core-i7 for 130 000 nodes and 800 000 edges.

[3] Maximal distance (L_1-norm) between the value of the node and the value of all its descendants.

References

1. Aho, A.V., Garey, M.R., Ullman, J.D.: The transitive reduction of a directed graph. SIAM J. Comput. 1(2), 131–137 (1972)
2. Angulo, J.: Geometric algebra colour image representations and derived total orderings for morphological operators–Part I: Colour quaternions. J. Vis. Commun. Image R. 21(1), 33–48 (2010)
3. Aptoula, E., Lefèvre, S.: A comparative study on multivariate mathematical morphology. Pattern Recogn. 40(11), 2914–2929 (2007)
4. 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)
5. Carlinet, E.: Extending the tree of shapes on colors. Master's thesis, ENS Cachan (2012)
6. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Thinnings, shortest path forests, and topological watersheds. IEEE Trans. Pattern Anal. Mach. Intell. 32(5), 925–939 (2010)
7. Goutsias, J., Heijmans, H.J.A.M., Sivakumar, K.: Morphological operators for image sequences. Comp. Vis. Imag. Under. 62(3), 326–346 (1995)
8. Heijmans, H.: Theoretical aspects of gray level morphology. IEEE Trans. Pattern Anal. Mach. Intell. 13(6), 568–592 (1991)
9. Jones, R.: Connected filtering and segmentation using component trees. Comp. Vis. Imag. Under. 75(3), 215–228 (1999)
10. Monasse, P., Guichard, F.: Scale-space from a level lines tree. J. Vis. Commun. Image R. 11(2), 224–236 (2000)
11. 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)
12. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Trans. Image Proc. 15(11), 3531–3539 (2006)
13. Passat, N., Naegel, B.: An extension of component-trees to partial orders. In: ICIP, pp. 3981–3984 (2009)
14. Passat, N., Naegel, B.: Component-trees and multivalued images: Structural properties. Technical report, INRIA-00611714 (2012), http://hal.inria.fr/inria-00611714
15. Ronse, C., Agnus, V.: Morphology on label images: Flat-type operators and connections. J. Math. Imaging Vis. 22(2), 283–307 (2005)
16. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. IEEE Trans. Image Proc. 9(4), 561–576 (2000)
17. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Trans. Image Proc. 7(4), 555–570 (1998)
18. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Trans. Pattern Anal. Mach. Intell. 30(7), 1132–1145 (2008)
19. 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)
20. Velasco-Forero, S., Angulo, J.: Supervised ordering in $\mathbb{R}^p$: Application to morphological processing of hyperspectral images. IEEE Trans. Image Proc. 20(11), 3301–3308 (2011)
21. 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 Trans. Pattern Anal. Mach. Intell. 30(10), 1800–1813 (2008)
22. Xu, Y., Géraud, T., Najman, L.: Morphological filtering in shape spaces: Applications using tree-based image representations. In: ICPR, pp. 485–488 (2012)

Inf-structuring Functions and Self-dual Marked Flattenings in bi-Heyting Algebra

Benjamin Perret

Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Equipe A3SI, ESIEE Paris,
2, boulevard Blaise Pascal
Cité Descartes BP 99
93162 Noisy le Grand CEDEX France
perretb@esiee.fr

Abstract. This paper introduces a generalization of self-dual marked flattenings defined in the lattice of mappings. This definition provides a way to associate a self-dual operator to every mapping that decomposes an element into sub-elements (i.e. gives a cover). Contrary to classical flattenings whose definition relies on the complemented structure of the powerset lattices, our approach uses the pseudo relative complement and supplement of the bi-Heyting algebra and a new notion of *inf-structuring functions* that provides a very general way to structure the space. We show that using an inf-structuring function based on connections allows to recover the original definition of marked flattenings and we provide, as an example, a simple inf-structuring function whose derived self-dual operator better preserves contrasts and does not introduce new pixel values.

Keywords: inf-structuring function, self-dual operator, flattening, Heyting algebra, connection, hyper-connection, image processing, mathematical morphology.

1 Introduction

Duality is a principle which states that the content of a numerical function remains the same after an inversion. Following this principle, an operator should process an image and its opposite symmetrically, *i.e.* be self-dual. Although this property is always achieved with linear filters, the problem is harder in mathematical morphology where the operations of infimum and supremum treat differently bright and dark objects.

Several authors have explored various manners to define self-dual morphological operators [3,4,9,5,20]. The first approaches are based on the notion of activity of a boolean operator which leads to the *activity lattice* [15] where given two dual binary operators, both their infimum (morphological centre) and supremum (flattening [7] and levelling [8,19]) produce self-dual operators. The second approaches rely on the notion of tree of shapes [9,5] which is the tree given by the relation of inclusion between the shapes (level lines) in the image. The definition of shapes and the inclusion relation being invariant to contrast inversion, this naturally leads to self-dual operators. A third approach consists in working with the module of the gradient of the function.

This paper generalizes the first approach and more precisely the marked flattening [18]. Marked flattenings are defined as increasing set operators which are thus

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 365–376, 2013.

easily extended to functions by thresholding and stacking. One calls the flattening $\theta_M(A)$ of the set A by the set M, the union of $\gamma_M(A)$, the set of connected components of A that intersect M, with the set of all pores (connected components of the background) that do not intersect M^c, i.e. those which are included in the marker: $\theta_M(A) = \gamma_M(A) \cup (A^c \cap \gamma^c_{M^c}(A))$.

Currently, the definition of the marked flattening is only feasible in sets due to the use of the complementation. We show that the use of the bi-Heyting algebra structure removes this limitation and lets us generalize the flattenings in order to use connections on functions [17]. In order to encompass all possible approaches for connections, in section 3, we define the new notion of *inf-structuring function* (isf), which is a mapping that associates to any function a set of smaller functions, and we propose two particular isfs. In section 4, we propose two swamping functions based on isfs and we show that it generalizes the classical notion of connected openings and closings. Then, section 5 presents the construction of the self-dual operator combining isf-swampings and Heyting algebra operators. We show that using adapted isfs, either based on connections or on hyper-connections, enables us to recover the original definition of flattenings and to define new self-dual operators that better preserve contrasts and that do not create new pixel values. Finally, we conclude the work in section 7 and we give a few perspectives.

2 Mathematical Preliminaries

Let $(\mathscr{L}, \vee_{\mathscr{L}}, \wedge_{\mathscr{L}}, \mathbf{0}, \mathbf{1}, \leq_{\mathscr{L}})$ be a complete lattice, where $\mathscr{L}$ is the set of elements of the lattice, $\wedge_{\mathscr{L}}$ (resp. $\vee_{\mathscr{L}}$) is the infimum (resp. supremum), $\mathbf{0}$ (resp. $\mathbf{1}$) is the smallest (resp. largest) element and $\leq_{\mathscr{L}}$ is the associated partial order. The lattice $\mathscr{L}$ is infinite $\vee$-distributive (resp. $\wedge$-distributive) if $\forall a \in \mathscr{L}$, $\forall B \subseteq \mathscr{L}$, we have $a \wedge (\bigvee_{b \in B} b) = \bigvee_{b \in B}(a \wedge b)$ (resp. $a \vee (\bigwedge_{b \in B} b) = \bigwedge_{b \in B}(a \vee b)$). It is infinite distributive if it is both infinite $\vee$- and $\wedge$-distributive. Given an element $a \in \mathscr{L}$ we note M_a (resp. M^a) the set of upper bounds of a (resp. lower bounds): $M^a = \{b \in \mathscr{L} \,|\, b \leq a\}$ and $M_a = \{b \in \mathscr{L} \,|\, b \geq a\}$

We also define the particular complete infinite distributive lattice of mappings from a non empty set $\mathscr{D}$ to a complete chain $\mathscr{T}$: $(\mathscr{F}, \vee_{\mathscr{F}}, \wedge_{\mathscr{F}}, \perp, \top, \leq_{\mathscr{F}})$, where $\mathscr{F}$ is the set of elements of the lattice, $\wedge_{\mathscr{F}}$ (resp. $\vee_{\mathscr{F}}$) is the pointwise infimum (resp. supremum) operator, $\perp$ (resp. $\top$) is the smallest (resp. largest) element and $\leq_{\mathscr{F}}$ is the associated partial order. For simplicity of notations, when possible, we omit the indices from the infimum, supremum and ordering symbol. In image processing, we usually define $\mathscr{T}$ as a closed subset of the completed real line $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ or $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, +\infty\}$.

2.1 Self-dual Operator

We give here a short presentation of self-duality, a deeper exploration is given in [4].

Definition 1. *An operator φ of $\mathscr{L}$ is an inversion if it is a decreasing involution (i.e. φ is a decreasing, bijective mapping from $\mathscr{L}$ to $\mathscr{L}$, such that $\forall a \in \mathscr{L}$, $\varphi^2(a) = a$).*

For simplicity of notation, we write a^* for an element $a \in \mathscr{L}$ where we assume that an inversion φ exists and that $a^* = \varphi(a)$. If the lattice $\mathscr{L}$ is distributive and if an inversion

φ exists for $\mathscr{L}$ then the image of $\mathscr{L}$ by φ is the dual lattice of $\mathscr{L}$ obtained by inverting the ordering relation and exchanging the infimum and supremum operators.

If $\mathscr{L} = \overline{\mathbb{R}}$ (resp. $\mathscr{L} = \overline{\mathbb{Z}}$), an inversion is given by: $\forall a \in \mathscr{L}$, $\varphi(a) = n - a$ with $n \in \mathbb{R}$ (resp. $n \in \mathbb{Z}$). If $\mathscr{L}$ is a closed subset $[n,m]$ of $\mathbb{R}$ or $\mathbb{Z}$, then an inversion is given by: $\forall a \in \mathscr{L}$, $\varphi(a) = n + m - a$. This extends to the lattice $\mathscr{F}$ using pointwise operations.

Definition 2. *Being given an operator α of $\mathscr{L}$, we define the dual operator α^* for φ by: $\forall a \in \mathscr{L}$, $\alpha^*(a) = (\alpha(a^*))^*$.*

The dual transformation of a is the inverse of the transformation of the inverse of a. Duality is a common method to define pairs of operators like openings and closings.

Definition 3. *An operator α is self-dual if it is equal to its dual operator: $\alpha = \alpha^*$.*

A self-dual operator treats an element and its inverse equally. In image processing this can be interpreted as a being covariant to contrast inversion. The convolution and the median filter are two well known examples of self-dual operators.

2.2 bi-Heyting Algebra

Heyting algebras are well known in the field of propositional logic, but, to our knowledge, has only been used by Stell et al. in the field of mathematical morphology [22,21].

Definition 4. *A complete Heyting algebra is a complete $\vee$-infinite distributive lattice $\mathscr{L}$ with a binary operator* pc *called relative pseudo-complement such that $\forall a,b \in \mathscr{L}$,* $\mathrm{pc}(a,b)$ *is the largest element such that $a \wedge \mathrm{pc}(a,b) \leq b$ [23].*

The lattice $\mathscr{F}$ is a Heyting algebra with $\forall f,g \in \mathscr{F}$, $\mathrm{pc}(f,g) = \bigvee \{h \,|\, f \wedge h \leq g\}$ (Fig. 1). If $\mathscr{L}$ is a complemented $\vee$-infinite distributive lattice, then the mapping $\mathrm{pc}(a,\mathbf{0})$ for $a \in \mathscr{L}$ is indeed the classical complementation.

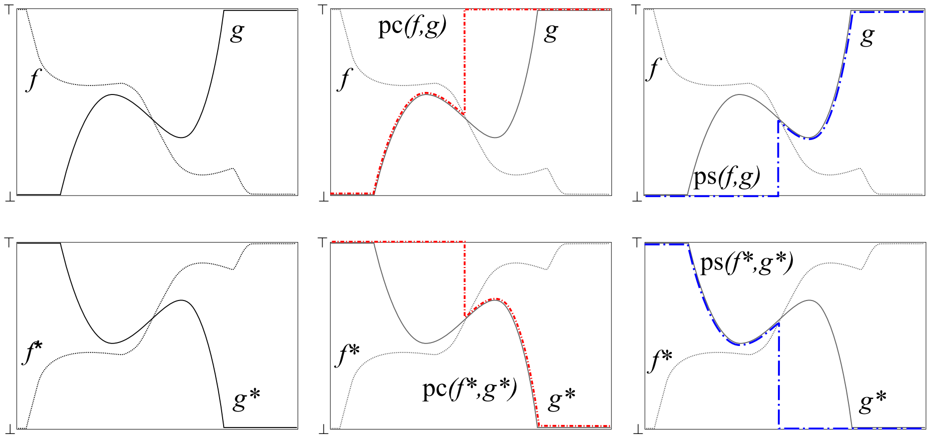


Fig. 1. Illustration of the relative pseudo-complement and the relative pseudo-supplement in $\mathscr{F}$. The first line shows the application of the relative pseudo-complement pc and the relative pseudo-supplement ps on two functions f and g. The second line shows the duality between pc and ps.

Definition 5. *A complete co-Heyting algebra is a complete $\wedge$-infinite distributive lattice $\mathscr{L}$ with a binary operator* ps*, called relative pseudo-supplement such that $\forall a,b \in \mathscr{L}$, $\mathrm{ps}(a,b)$ is the smallest element such that $a \vee \mathrm{ps}(a,b) \geq b$ [6].*

The lattice $\mathscr{F}$ is a co-Heyting algebra with $\forall f,g \in \mathscr{F}$, $\mathrm{ps}(f,g) = \bigwedge \{h \,|\, f \vee h \geq g\}$.

An algebra that is both a Heyting and a co-Heyting algebra is then referred as a *bi-Heyting algebra* [12].

ps and pc are dual operators: $\forall a,b \in \mathscr{L}$, $\mathrm{ps}(a,b) = (\mathrm{pc}(a^*,b^*))^*$ and conversely $\mathrm{pc}(a,b) = (\mathrm{ps}(a^*,b^*))^*$. Fig. 1 shows two examples of applications of pc and ps in $\mathscr{F}$. We see that $\mathrm{pc}(f,g)$ is equal to $\top$ where f is smaller than g and to g otherwise, conversely $\mathrm{ps}(f,g)$ is equal to $\perp$ where f is larger than g and to g otherwise.

3 Inf-structuring Functions

In this section, we present the new concept of isf which is a very general type of mapping that associates to each element of a lattice a set of sub-elements (Fig. 2)

Definition 6. *We say that* $\mathrm{s} : \mathscr{F} \mapsto \mathscr{P}(\mathscr{F})$ *is an inf-structuring function (isf) of* $\mathscr{F}$ *if:*

1. *$\forall f \in \mathscr{F}$, $\mathrm{s}(f) \subseteq M^f$ (i.e. $\forall g \in \mathrm{s}(f), g \leq f$): all sub-elements are smaller than f.*
2. *$\forall f \in \mathscr{F}$, $\bigvee \mathrm{s}(f) = f$: the supremum of the sub-elements of f is equal to f*

One can note that an important difference with (hyper-)connections is that an isf can decompose an element into comparable elements.

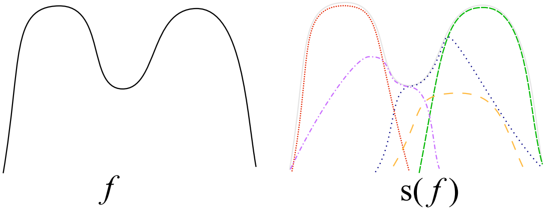


Fig. 2. Example of decomposition of a function f into a set of five lower functions $\mathrm{s}(f)$. The assumptions made on the content of $\mathrm{s}(f)$ are very weak.

A simple way of constructing isfs is to use the tool-box from the connection theory. We present here two isfs, the first one based on the set connections [16] and the second one the hyper-connections [17].

3.1 Set Connections for Isf

Set connections are a convenient way to describe how elements of a set are grouped in so called connected components [16]. Being given a connection $\mathscr{C}$ on the powerset lattice $\mathscr{P}(E)$, a subset A of E and a point x in E, the connected component of A containing

x (for $\mathscr{C}$), noted $\gamma_x(A)$, is the largest element of $\mathscr{C}$ containing x and included in A. We can now define the isf based on the connection $\mathscr{C}$ and denoted by $\mathrm{s}_{\mathscr{C}}$ as:

$$\forall f \in \mathscr{F},\ \mathrm{s}_{\mathscr{C}}(f) = \left\{ \bigvee_{v \leq t} \mathrm{cyl}\left(\gamma_x\left(\overline{f}^{v} \right), v \right) \,\middle|\, t \in \mathscr{T},\ x \in \mathscr{D} \right\} \tag{1}$$

where $\mathrm{cyl}(X,v)$ represents the cylinder of base X and level v: for all $x \in \mathscr{D}$, $\mathrm{cyl}(X,v)(x)$ equals v if $x \in X$ and $\mathbf{0}$ otherwise. And, $\overline{f}^{t}$ is the threshold of the function $f \in \mathscr{F}$ at level $t \in \mathscr{T}$, i.e. the set of points where the value of f is larger than t: $\overline{f}^{t} = \{x \in \mathscr{D} \,|\, f(x) \geq t\}$.

This construction, depicted in Fig. 3, is indeed closely linked (see Prop. 1) to the definition of anti-extensive connected operators in gray-level images [14].

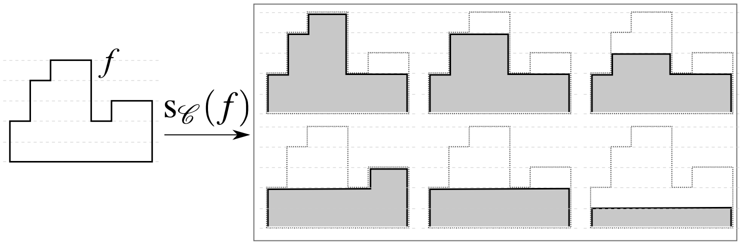


Fig. 3. Set connection based isf: the function on the left is decomposed into the 6 functions in the right box. We assume here that the value domain is discrete (dashed horizontal lines).

3.2 Ultimate Flat Zone Isf

We define here the *ultimate flat zone* isf noted s_{ufz}. Each element of the isf of f corresponds to the infimum between a maximal flat function of f (*i.e.* a flat function with connected support such that there do not exist another flat function strictly larger than it and lower than f) and a flat zone of f (Fig. 4). This construction can be formalized with hyper-connections [17] as an iterative decomposition into z-zones [1,11] using the h-connection of flat functions [10].

4 Isf Swampings

The definition of the generalized flattening is done in two steps. We first give a generalized definition of a swamping (marked reconstruction) based on the notion of isf. Then, in the next section, we define the generalized flattening using the swamping function and the bi-Heyting algebra.

An isf provides a first way to structure the space by considering a notion of local minima conditionally to the decomposition. We define an elementary swamping function, i.e. a marked reconstruction, $\beta : \mathscr{F} \times \mathscr{F} \mapsto \mathscr{F}$ by:

$$\beta(f,m) = \bigvee \min(M_m \cap \mathrm{s}(f)) \tag{2}$$

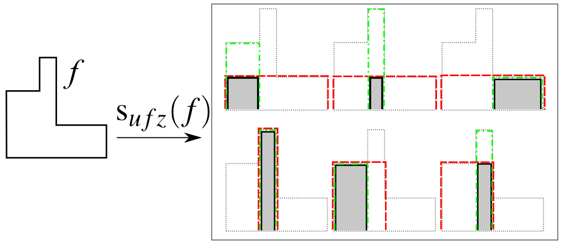


Fig. 4. Ultimate flat zone isf: the function on the left is decomposed into 6 functions showed in the right box. Each element of the isf is given by the infimum between a maximal flat function (red dashed line) and a flat zone (green dot-dash line).

where f is the processed element, m is the marker and $\min(X)$ is the set of minimal elements of the set X ($\min(X) = \{x \in X \,|\, \forall y \in X,\ y \leq x \Rightarrow y = x\}$). The use of the min in this formula insures that for each element g of $\mathrm{s}(f)$ there exists a marker m such that $g = \beta(f,m)$.

Then, we define the isf-swamping $\alpha : \mathscr{F} \times \mathscr{F} \mapsto \mathscr{F}$ as an extension of β:

$$\forall f,m \in \mathscr{L},\ \alpha(f,m) = \bigvee_{n \leq m} \beta(f,n) \tag{3}$$

The operators α and β and their differences are illustrated in Fig. 5. Observe that the purple sub-element (dot-dash line) is included in $\alpha(f,m)$ but not in $\beta(f,m)$ as it only intersects m. Also, in both cases, the blue sub-element (largely spaced dotted line) that also intersects m is not included because there does not exist a $n \in M^m$ such that n is smaller than this sub-element but larger than the orange one (spaced dashed line).

The behaviour of α and its dual operator are illustrated in Fig. 6. α is similar to a rasing of f marked by m as it removes all sub-elements of a that are completely outside the marker. On the other side, the dual operator α^* acts as a flooding of f marked by m, adding all sub-elements of the inverse that are completely included in the marker.

α is not increasing with respect to f (Fig. 7) but it is trivially increasing with respect to m as more and more sub-elements are included in the result when m increases. The fact that α is not increasing with respect to its first argument is not a real drawback as this property is indeed really important only for binary operators in order to naturally

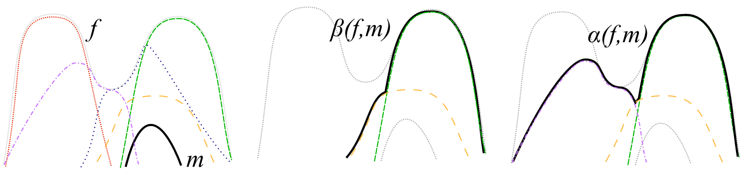


Fig. 5. Example of application of the β and α operators on the function f decomposed in $\mathrm{s}(f)$ and marked by m. Whereas β selects the lowest sub-elements that are greater than the marker, α selects the lowest sub-elements whose support intersects the support of the marker.

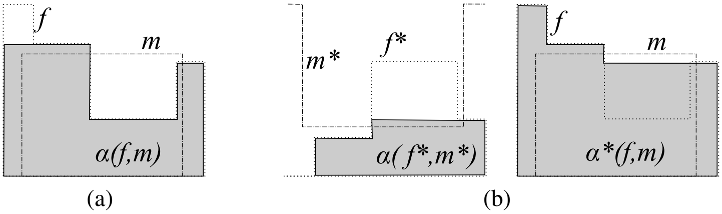


Fig. 6. Illustration of the α operator and its dual operator with the *ultimate flat zone* isf (Fig. 4). Fig. (a) shows a function f (dotted line) a marker m (dot-dash line) and the result $\alpha(f,m)$ (in grey). The first image of Fig. (b) shows the result of $\alpha(f^*,m^*)$. Finally, the last image shows the result of the dual operator $\alpha^*(f,m)$.

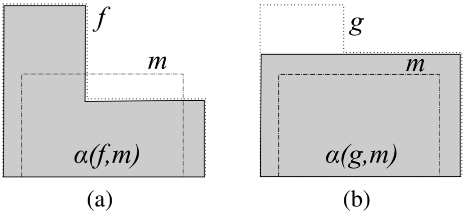


Fig. 7. Example showing that α is generally not increasing with the *ultimate flat zone* isf (Fig. 4). Fig. (a) shows the result of $\alpha(f,m)$, whereas we have the result of $\alpha(g,m)$ on Fig. (b). We have $f \leq g$ but $\alpha(f,m)$ and $\alpha(g,m)$ are not comparable.

extend them to the grayscale case by stacking. It is clearly anti-extensive with respect to f but not to m. It is generally not idempotent with respect to f.

Proposition 1. *When the isf is based on connection* ($\mathrm{s}_{\mathscr{C}}$*: Eq. 1), then* α *reduces to a classical marked connected opening (and closing for* α^**).*

5 Generalized Flattenings

Based on α, we can construct a new self-dual operator $\Theta : \mathscr{L} \times \mathscr{L} \mapsto \mathscr{L}$ following the method used for the definition of binary flattenings [7,18]. Nevertheless, binary flattenings rely on the complemented structure of the binary lattice and we will show that this construction can be extended using the notion of relative pseudo-complement and -supplement. Formally, Θ is defined by:

$$\forall f,m \in \mathscr{L},\ \Theta(f,m) = \alpha(f,m) \vee \mathrm{ps}(f,\alpha^*(f,m)). \tag{4}$$

This equation is illustrated in Fig. 8. Its first part: $\alpha(f,m)$ takes the supremum of the smallest sub-elements of f that intersect m. Conversely, $\alpha^*(f,m)$ takes the inverse of the supremum of the smallest sub-elements of f^* that intersect m^* which can be interpreted as adding to f the "parts" of f^* that are completely included in m.

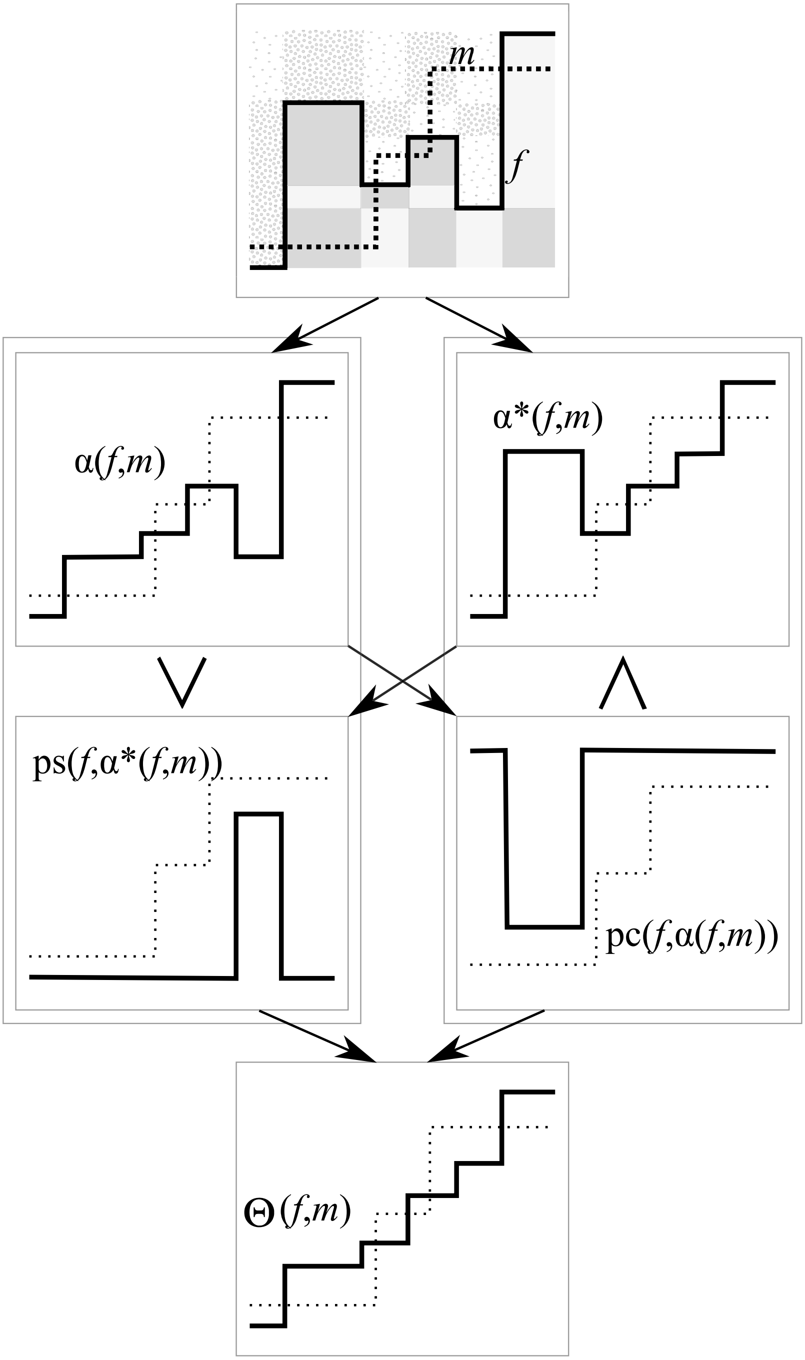


Fig. 8. Example of application of the operator $\Theta(f,m) = \alpha(f,m) \vee \mathrm{ps}(f,\alpha^*(f,m))$ (Eq. 4) and its equivalent definition $\Theta(f,m) = \alpha^*(f,m) \wedge \mathrm{pc}(f,\alpha(f,m))$ (Prop. 2) using an ultimate flat zone isf (Fig. 4).

Then $\mathrm{ps}(f,\alpha^*(f,m))$ filters $\alpha^*(f,m)$ in order to only keep those added parts. Finally, $\Theta(f,m)$ is the supremum between the filtered version $\alpha(f,m)$ and the added parts from $\alpha^*(f,m)$. Intuitively the final result is composed of the parts of f that intersect m and the parts of f^* that are below m.

Proposition 2. *For all $f,m \in \mathscr{L}$, we have* $\Theta(f,m) = \alpha^*(f,m) \wedge \mathrm{pc}(f,\alpha(f,m))$

This equation is illustrated in Fig. 8. Here, $\alpha^*(f,m)$ adds to f the "parts" of f^* that are completely included in m. On the other side, $\mathrm{pc}(f,\alpha(f,m))$ determines which parts of f are not selected by $\alpha(f,m)$, i.e. those that do not intersect m. Finally we remove from $\alpha^*(f,m)$ the parts extracted by $\mathrm{pc}(f,\alpha(f,m))$.

Proposition 3. Θ *is an self-dual operator:* $\Theta(f,m) = \Theta(f^*,m^*)^*$

This is an immediate consequence of Prop. 2 and the duality between pc and ps.

Proposition 4. *When the isf is based on connection* ($\mathrm{s}_{\mathscr{C}}$*: Eq. 1), then* Θ *reduces to the classical connected marked flattening operator as defined in [18].*

This is a direct consequence of Prop. 1.

5.1 Discussion

It is also possible to interpret Θ as the supremum of activity between α and α^* which corresponds to the fact that binary marked flattenings can themselves be defined as the supremum of activity of two operators. Nevertheless, this previous definition was only feasible in complemented lattices while it is here directly expressed in the lattice of mappings thanks to the bi-Heyting algebra structure. One can note that Serra touched upon this construction with the Lemma 8.2 of [15] where he noticed that the use of an infinite-distributive lattice was usefull in this frame.

One can note that Θ is not a generalization of marked levellings of [19] because the later rely on two particular relations. In [19] a component of the foreground is selected if it *touches* the marker and at the opposite a component of the background is selected if it is *strictly included* in the marker. While in our definition, a part of the function is selected if it *intersects* the marker and at the opposite a part of the inverse of the function is selected it is *under* the marker. While the relations used in [19] ensure that the resulting operator is adjacency stable we do not have this property and Θ is thus not a levelling [2]. Nevertheless, if we use two different markers in the definition of Θ, then, we can recover the classical marked leveling by taking the dilation of m as the marker of α and the erosion of m as the marker of α^*.

One can also note that the self-duality of the complete application of the operator (computation of m and application of Θ) is indeed only verified if the marker is computed with a self-dual operator. The question of the real properties of morphological operators based on context information (marker for flattenings and levelings, set of structural elements for spatially variant morphology) has been discussed by several authors [13,19] and more extensively in [2].

6 Applications

We propose applications to exhibit some differences between the self-dual operator Θ obtained with the *ultimate flat zone* isf and the classical flattenings or levelings.

The first line of Fig. 9 illustrates the fact that Θ cannot introduce new pixel values with the *ultimate flat zone* isf. The original image is a simple chess board while the marker is the result of a convolution of the chess board with a Gaussian kernel. With the classical leveling new pixel values coming from the marker are introduced in the result while in our new definition, the result is equal to the original image. The second line of Fig. 9 relies this time on a noisy version of the chess board image. Similarly to the classical leveling, our operator reduces drastically the noise level, does not move frontiers and moreover, it also better preserves the contrast. A profile view of the same example is given Fig. 10.

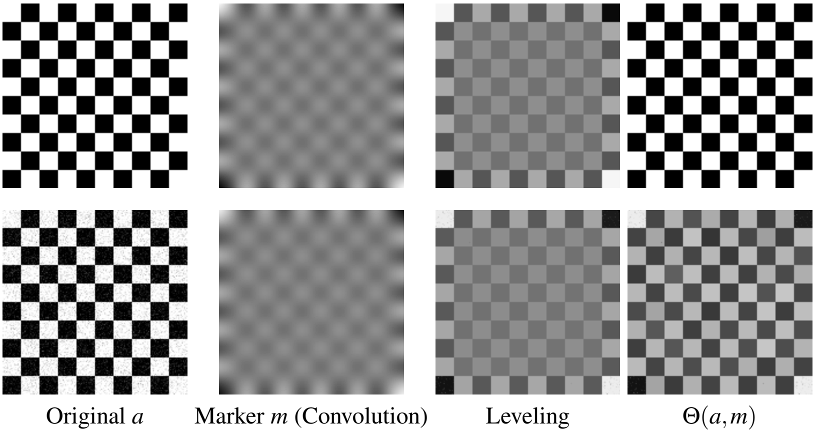


Fig. 9. Comparison of the generalized flattening Θ with the classical levelling. First line: the original image does not contain any noise, we see here that the use of the *ultimate flat zone* isf prevents the introduction of new grey levels in the result. Second line: here the original image is noisy and we show that Θ better preserves the contrast as it tends to reconstruct the maxima.

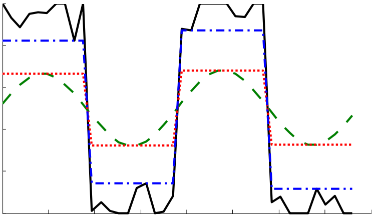
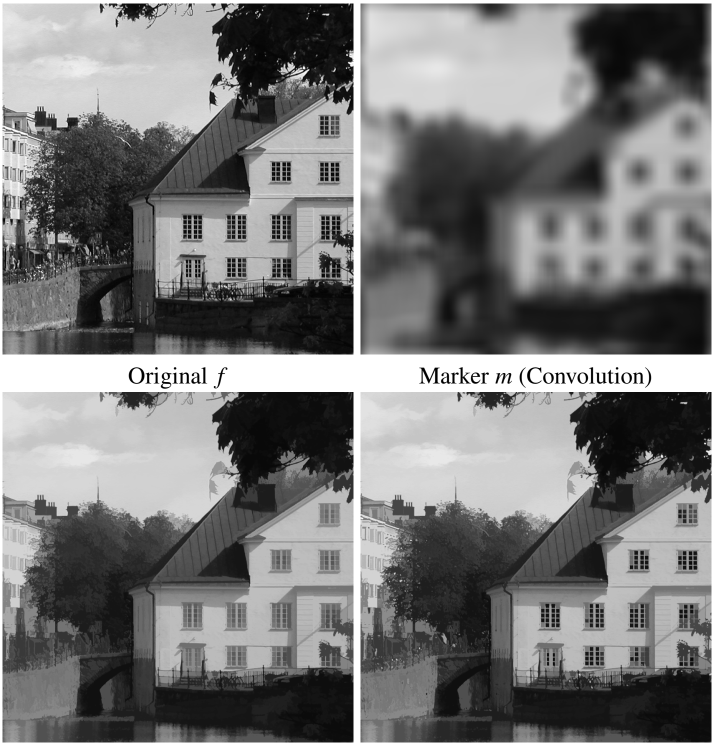


Fig. 10. Illustration of a profile through the images of the second line of Fig. 9. The black solid curve is the original image, the green dashed line is the marker, the red dotted line is the classical leveling and the blue dotted-dashed line is the result of Θ (built upon the *ultimate flat zone* isf). We see that Θ tends to reconstruct the maxima and thus better preserves the contrast.



Original f

Marker m (Convolution)

Marked leveling

$\Theta(f,m)$

Fig. 11. Comparison of Θ (built upon the *ultimate flat zone* isf) with to the connected leveling.

Finally, Fig. 11 shows a real life application on a picture of Uppsala. One can observe that our operator performs a simplification of the image that is similar to the one provided by the connected leveling while offering a better preservation of the contrast.

7 Conclusion

In this article we have proposed a generalization of the self-dual marked flattenings that provides a way to associate a self-dual operator to every mapping that decomposes an element into sub-elements. It is defined directly in the lattice of mappings thanks to its bi-Heyting algebra structure and it relies on the new notion of *inf-structuring functions*. We derive two new swampings from the notion of isf and we show that they generalize the connected openings. Then, we define a new self-dual operator based on those swampings that generalizes the marked flattenings. Finally, we show that using a *naive* isf allows to better preserve the contrast, which suggests that even better self-dual operators can be obtained through the definition of more sophisticated isfs.

In future work, we plan to explore two directions: first the possibility to develop the notion of activity based on the bi-Heyting algebra structure and, second, the deep exploration of the notion of *inf-structuring functions* and more precisely its various links with connections, hyper-connections and the associated (h-)connected operators.

Acknowledgements. I would like to thank Prof. Jean Serra for his valuable comments and corrections on the initial version of this paper.

References

1. Braga-Neto, U., Goutsias, J.: A theoretical tour of connectivity in image processing and analysis. JMIV 19(1), 5–31 (2003)
2. Crespo, J.: Adjacency stable connected operators and set levelings. IVC 28(10), 1483–1490 (2010)
3. Heijmans, H.J.A.M.: Self-dual morphological operators and filters. JMIV 6, 15–36 (1996)
4. Heijmans, H.J.A.M., Keshet, R.: Inf-semilattice approach to self-dual morphology. JMIV 17(1), 55–80 (2002)
5. Keshet, R.: Shape-tree semilattice. JMIV 22(2-3), 309–331 (2005)
6. Lawvere, F.W.: Intrinsic co-heyting boundaries and the leiniz rule in certain toposes. In: Category Theory. LNM, vol. 1488, pp. 279–281. Springer (1991)
7. Meyer, F.: From connected operators to levelings. In: Mathematical Morphology and its Applications to Image and Signal Processing, pp. 191–198. Kluwer (1998)
8. Meyer, F.: The levelings. In: Mathematical Morphology and its Applications to Image and Signal Processing, pp. 199–206. Kluwer (1998)
9. Monasse, P., Guichard, F.: Fast computation of a contrast-invariant image representation. IEEE TIP 9(5), 860–872 (2000)
10. Perret, B., Lefèvre, S., Collet, C.: Toward a new axiomatic for hyper-connections. In: Soille, P., Pesaresi, M., Ouzounis, G.K. (eds.) ISMM 2011. LNCS, vol. 6671, pp. 85–95. Springer, Heidelberg (2011)
11. Perret, B., Lefèvre, S., Collet, C., Slezak, É.: Hyperconnections and hierarchical representations for grayscale and multiband image processing. IEEE TIP 21(1), 14–27 (2012)
12. Reyes, G.E., Zolfaghari, H.: Bi-heyting algebras, toposes and modalities. Journal of Philosophical Logic 25, 25–43 (1996)
13. Roerdink, J.B.T.M.: Adaptivity and group invariance in mathematical morphology. In: IEEE ICIP 2009, pp. 2253–2256 (2009)
14. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE TIP 7(4), 555–570 (1998)
15. Serra, J.: The centre and self-dual filtering. In: Image Analysis and Mathematical Morphology. II: Theoretical Advances, pp. 159–180. Academic Press (1988)
16. Serra, J.: Mathematical morphology for boolean lattices. In: Image Analysis and Mathematical Morphology. II: Theoretical Advances, pp. 37–58. Academic Press (1988)
17. Serra, J.: Connectivity on complete lattices. JMIV 9(3), 231–251 (1998)
18. Serra, J.: Connections for sets and functions. Fundam. Inform. 41, 147–186 (2000)
19. Serra, J., Vachier, C., Meye, F.: Levelings. In: Mathematical Morphology, pp. 199–228. ISTE and John Wiley & Sons (2010)
20. Soille, P.: Beyond self-duality in morphological image analysis. IVC 23, 249–257 (2005)
21. Stell, J.G.: Relations in mathematical morphology with applications to graphs and rough sets. In: Winter, S., Duckham, M., Kulik, L., Kuipers, B. (eds.) COSIT 2007. LNCS, vol. 4736, pp. 438–454. Springer, Heidelberg (2007)
22. Stell, J.G., Worboys, M.F.: The algebraic structure of sets of regions. In: Frank, A.U. (ed.) COSIT 1997. LNCS, vol. 1329, pp. 163–174. Springer, Heidelberg (1997)
23. Vickers, S.: Topology via Logic. Cambridge University Press (1989)

From Extrema Relationships to Image Simplification Using Non-flat Structuring Functions

Guilherme Polo and Neucimar J. Leite

Institute of Computing, University of Campinas, UNICAMP
13083-852 Campinas, SP, Brazil
{ggpolo,neucimar}@ic.unicamp.br

Abstract. Image simplification plays a fundamental role in Image Processing to improve results in complex tasks such as segmentation. The field of Mathematical Morphology (MM) itself has established many ways to perform such improvements. In this paper, we present a new approach for image simplification which takes into account erosion and dilation from MM. The proposed method is not self-dual and only single-band signals under a discrete domain are considered. Our main focus is on the creation of concave structuring functions based on a relation between signal extrema. This relation is given by two extrema according to their degree of separation (distance) and the respective heights (contrast). From these features, a total order relation is produced, thus supplying a way to progressively simplify the signal. Some two-dimensional images are considered here to illustrate in practice this simplification behavior.

Keywords: Mathematical Morphology, Regional Extrema, Concave Structuring Functions, Image Simplification.

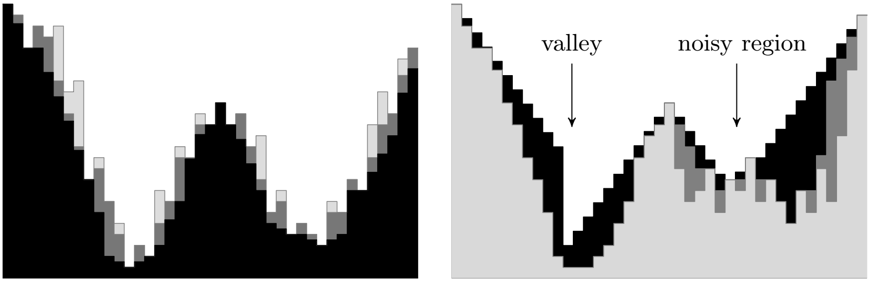
1 Introduction

Signal simplification involves reduction and/or improvement of the provided data. Ideally, one would want to remove all the irrelevant data for the task being performed while preserving everything else. In the case of edge recovery, for instance, it is also needed to improve the input signal. In general, simplifying an image is characterized as an inverse problem with possibly many solutions, thus it is an ill-posed task. Nonetheless, many specific and general methods have been developed as it is indispensable for an effective segmentation, for example. Here we present a general approach based on erosion/dilation from Mathematical Morphology (MM) [12], with non-flat scaled structuring functions. No particular connectivity is assumed, and the method is described in terms of graphs.

The starting point of our approach is the creation of an order relation $\mathcal{R}$ taking into account the signal extrema (either maxima or minima). The anticipated benefit of $\mathcal{R}$ is twofold: a) the details of the used structuring function are made unimportant for the user, b) the order conveys a clear indication for the degree of simplification. The first item is particularly important because it is often

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 377–389, 2013.

the case [7,4] that the scaling of the structuring function is closely related to its shape, and arbitrarily selecting a scale does not tell much about how it will affect the signal. The specificities on how $\mathcal{R}$ is constructed and the corresponding simplification process are discussed in the following sections. Figure 1 provides a general view of the method which is not self-dual, so it is possible to opt to work with either regional minima or maxima. Figure 1a can be seen as two valleys corrupted by noise whose simplification process manages to suppress. The original signal in Fig. 1b shows a valley next to a noisy minimum region. In this example, the simplification is obtained by suppressing regional maxima of the signal which indirectly suppresses minima of similar height in the originally noisy region. In both cases, note the ramp-like path created between extrema.



(a) Example considering regional minima. The whole area describes the input signal, and the darker the area the greater the simplification.

(b) Example considering regional maxima. The lighter area describes the input signal. The darker areas together with the lighter ones represent the increasing simplification.

Fig. 1. Overview of the simplification process

To situate this work, some examples of morphological approaches to signal simplification are considered. On one hand, the proposed method is related to those that provide means for simplification by considering specific signal features. This includes the contrast measure bestowed by dynamics [5,1] and other works [15,14] that select signal features based on the extinction values (e.g., area or volume), as we also measure the persistence of the image extrema. In addition, we simplify a signal, as illustrated in Fig.1, by using these obtained measures. On the other hand, simplifications based on connected operators [13,9] are also related to the proposed approach, although our focus is on merging flat zones without extending them. In terms of relief and unlike flooding, our transformation can be seen as a producer of geomorphological benches as commonly found in open-pit mines. This difference impacts the simplification, since a final flooding can be seen as a more abrupt process while ours defines progressive steps represented by ramps.

The next section begins by quickly revisiting restrictions upon our considered structuring functions. In Sects. 3-5 the simplification procedure is detailed in a more formal manner. Section 6 illustrates the simplification process on some two-dimensional images. Finally, conclusions are drawn in Sect. 7 which also discusses future works on this subject.

2 Morphology with Concave Functions

Given the importance of the morphological operations of erosion, ε, and dilation, δ, of a function f with a structuring function (SF) g, we remember them, respectively, as in [12]: $\varepsilon(f,g)(x) = \inf_{-y \in g} f(x-y) - g(y)$; $\delta(f,g)(x) = \sup_{y \in g} f(x-y) + g(y)$. Here, the interest is in working with SFs that preserve certain characteristics. From [12] we have that both non-flat erosion and dilation maintain the increasing property on f when g contains the origin. So, in this work, g is restricted to always contain the origin. Furthermore, dilation with g is increasing with respect to itself while erosion with SFs g_1 and g_2 yields the following property:

$$g_1 \leq g_2 \Rightarrow \varepsilon(f, g_1) \geq \varepsilon(f, g_2), \tag{1}$$

and, therefore, erosion is decreasing with respect to g. Next, other restrictions related to our SF g are discussed.

In special, recall the link of erosion and dilation with the order-statistics filters [11]. For a flat SF g, $\varepsilon(f,g)(v) = x$ corresponds to the $1st$ ranked-order filter (rank 1), while dilation is given by the $\#N_v$-th ranked-order filter. The notation $\#X$ stands for the cardinality of set X, and N_v corresponds to the neighbors of v, including v itself. Now, consider a non-flat g. We start by reusing the definition of weighted rank-order filters in [16]. This weighted rank (*wrank*) applies the weights in g to $f(u \in N_v)$ before ordering and picking an element. Thus, *wrank* produces the $1st$ wranked-order filter (wrank 1) for $\varepsilon(f,g)(v)$ and the corresponding $\#N_v$-th wranked-order filter for $\delta(f,g)(v)$. In the following, a stronger condition for g is sought so that

$$\text{rank } 1 \leq \text{ wrank } 1 \leq \text{ wrank } \#N_v \leq \text{ rank } \#N_v. \tag{2}$$

In this sense, it is clear that our sought-after g is in the domain of the order statistics and that, in general, a free-form g does not respect the ordering given by (2). To this end, concave SFs, also known as concave down in literature, are considered.

Some authors [7,3,4] have approached the use of these concave SFs. The work in [3] focused on parabolic-shaped functions defined by $g_t(x) = -\frac{1}{2t}x^2$, where $t > 0$ is a scale parameter. Besides the separability of the parabolic g_t, it assumes $\sup(g_t) = 0$ and $g_t(0) = 0$. In [7], it was considered that when $\sup(g_t) = 0$, level-shifting effects are avoided, as well as horizontal translations when $g_t(0) = 0$. Still in [7], it was proved that the scale-space [8] conditions of non-enhancement of extrema and causality (non-creation of extrema) hold for such SFs. Since these characteristics are also well desired for signal simplification, our g is further restricted to these assumptions, albeit in discrete form, in order to respect the corresponding properties. As a consequence of the scale-space properties, it is easy to see that the order in (2) is maintained. Finally, consider the height of the support of g as $h_g = |\inf(g)|$. Equation (1) is then extended to

$$(g_1 \leq g_2 \iff h_{g_1} \geq h_{g_2}) \Rightarrow \varepsilon(f, g_1) \geq \varepsilon(f, g_2). \tag{3}$$

With these considerations, the proposed simplification can now be introduced.

3 Extrema Relationship

Let $f : \mathbb{Z} \times \mathbb{Z} \to \mathbb{R}$ be a signal described by a connected labeled simple graph $G = (V, E, f)$. The set of edges $E \subseteq V \times V$ outlines the connectedness of the underlying signal over its domain of vertices V. We consider that a given signal f is devoid of structure which is provided by G. Given this configuration, we claim to construct a morphological operator that simplifies f by means of the suppression of regional extrema. As will be seen elsewhere, the corresponding transformation operates by effectively creating ramps between these extrema.

The main concern of our approach is the relationship among the extrema of the signal f. From this relation, structuring functions are derived such that they guarantee a well-behaved suppression of extrema in the sense that no new extrema are introduced in the transformed signal. Before stating how these extrema are related, let us consider the following definitions.

Definition 1. *Let $G = (V, E)$ be a graph. Let $u, v \in V$. Let P be the set of vertices along one of the paths of shortest length between u and v. Then u and v are l'-*separated *according to $l'(u, v) = \#(P - \{u, v\})$.*

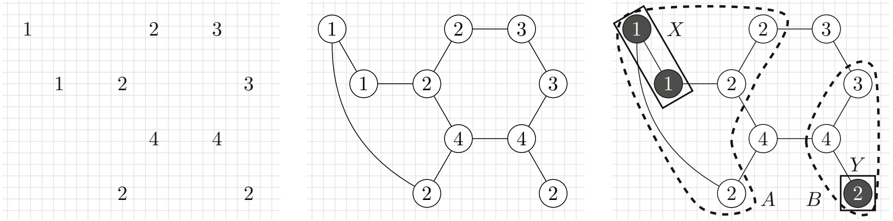
Definition 2. *Let $G = (V, E)$ be a graph. Let $E_A \subseteq E$ and $E_B \subseteq E$ be two subsets from which the sets A and B are built such that A contains all the vertices that form the edges in E_A, respectively for B, and respect $A \cap B = \varnothing$. Then A and B are l-*separated *by $l(A, B) = \min\{l'(u, v) \,|\, \forall u \in A, \forall v \in B\}$.*

From Definition 1, it is established that given two distinct regional extrema X and Y (both maxima or minima) and any $u \in X$, $v \in Y$, then $l'(u, v) \geq 1$. This result, together with Definition 2, yields $l(X, Y) \geq 1$. From now on, we will deal exclusively with sets and shorten the notation to just l (if context allows) which indicates the minimal number of vertices standing between both involved sets. Next, sets are treated as being neighbors if they respect the following definition.

Definition 3. *Given the same conditions as in Definition 2, then A and B are* s-neighbors *if $l(A, B) = 1$.*

In particular, by Definition 2, an edge (u, v) in G produces $l(\{u\}, \{v\}) = 0$ and, therefore, sets describing adjacent vertices are not *s-neighbors*. Figure 2 exemplifies the definitions discussed so far. The signal on Fig. 2a is structured according to some arbitrary graph shown in Fig. 2b. Based on it, the annotations of Fig. 2c are built. In this figure there are two regional minima, X and Y

(the solid rectangles), with $l\,(X,Y) = 3$. Now, suppose a function z that produces the sets A, B from X, Y (the closed dashed curves), such that $X \subseteq A$ and $Y \subseteq B$ with $l(z(X,Y)) = l(A,B) = 1$. In Mathematical Morphology, one such z would be the watershed transform described as "Meyer 2" [6], where the vertices labeled 3 and 4 in Fig. 2c, outside of both sets A and B, act as water divisors.



(a) Unstructured signal f on a discrete space.

(b) A graph G provides the structure for f.

(c) Minima X, Y, and the built *s-neighbor* sets A, B.

Fig. 2. Illustrations of some developed considerations and definitions

The above mentioned z function has an important role in our simplification method since it is useful to define, based on the produced sets A and B, whether two minima (maxima) are *s-neighbors.* Note that the use of such a function reduces the amount of relations generated among extrema. In the following, we assume the use of a z function that relates every pair of regional minima (maxima) as *s-neighbors.*

From these observations, the proposed relation among signal extrema can be introduced. Consider two extrema X and Y (both minima or maxima) from a signal f, and $u \in X$, $v \in Y$. If there is a function z so that the sets given by $z(X,Y)$ are *s-neighbors*, then the following equation defines the height of the scaled structuring function used in the simplification process:

$$\sigma = \frac{|f(u) - f(v)|}{l(X,Y)}. \tag{4}$$

If (4) is considered for every pair of *s-neighbor* sets, one then obtain a meaningful collection of values that can be used in the suppression of the signal extrema. Now, by grouping each height σ with $l(X,Y)$ in a tuple, a strict total order relation can be defined as follows. Let $S = \{(\sigma_1, l_1), \ldots, (\sigma_n, l_n)\}$ represent the set of these tuples. A relation $\mathcal{R}$ on S, between two tuples, is given by:

$$(\sigma_p, l_p) \prec (\sigma_o, l_o) \iff \sigma_p > \sigma_o \lor (\sigma_p = \sigma_o \land l_p < l_o). \tag{5}$$

The reasoning for such an ordering is closely related to the way the simplification occurs, as explained next.

4 Simplification Activity

The simplification process starts by choosing a tuple $t = (\sigma_t, l_t) \in S$, from which the σ_t value is used to define the structuring function $g_t : \mathbb{Z} \times \mathbb{Z} \to \mathbb{R}$. This function is connected according to the graph of the input signal f and its size extends only to the neighbors of a vertex v. The shape of g_t is given by:

$$g_t(x) = \begin{cases} 0 & \text{if } x = 0, \\ -\sigma_t & \text{otherwise.} \end{cases} \tag{6}$$

Note that g_t respects $\sup(g_t) = 0$ and $g_t(0) = 0$. Furthermore, it is a concave function, as discussed in Sect. 2, and has no problem in extending itself to handle varying neighborhoods of the underlying graph. In a rectangular 8-connected grid, g_t is a discretized pyramid. By using this structuring function, the signal can be simplified by either performing erosions (for regional minima) or dilations (for regional maxima), as further discussed in the paper.

Let $\mathit{\Gamma}(f, t)$ be the function that simplifies the signal f with the tuple $t \in S$. This implies the construction of a function g from t, as in (6), written from now on as g_t. Also, the first element in t is referred as σ_t, and the second as l_t. In such a case, $\mathit{\Gamma}(f, t)$ performs l_t erosions/dilations, using g_t as structuring function.

The meaning of relation $\mathcal{R}$ in (5) can be properly understood for regional minima (and by duality for regional maxima) as follows. Let f_p and f_o be the result of $\mathit{\Gamma}(f, p)$ and $\mathit{\Gamma}(f, o)$, respectively. First, suppose the situation in which the height of the corresponding structuring functions is the same, i.e., $\sigma_p = \sigma_o$:

i) If $l_p > l_o$, then $f_p \leq f_o$ due to the increasing and anti-extensivity properties of erosion with g_p and g_o;
ii) If $l_p < l_o$, then $f_p \geq f_o$, according to these same properties.

These simple considerations provide the basis for the ordering given by (5). In the earlier case, (i), we have $f - f_p \geq f - f_o$. Thus, opting for the second case, an ordering of non-decreasing activity [10] is established in this situation.

Now, let us consider $\sigma_p > \sigma_o$:

iii) When $l_p = l_o$:
  a) For $l_p = l_o = 1$, the property in (3) gives $f_p \geq f_o$
  b) The general case for $l_p = l_o = n$ can be demonstrated by induction using the anti-extensivity property of erosion together with (3).
iv) If $l_p < l_o$, then the first l_p erosions yields the result in (iii.b), and we are left with $f'_p = \mathit{\Gamma}(f, p) = f_p$ and $f'_o = \mathit{\Gamma}(f, (\sigma_o, l_p))$. Now, there are $l_o - l_p$ remaining erosions which are executed exclusively on f'_o with g_o and, hence, $f_p > f_o$.
v) Finally, if $l_p > l_o$, then a more subtle situation is encountered. Carrying out l_o erosions on f produces, as before, the result in (iii.b), which in turn means $f'_p = \mathit{\Gamma}(f, (\sigma_p, l_o))$ and $f'_o = \mathit{\Gamma}(f, o) = f_o$. Now, the remaining $l_p - l_o$ erosions are to be done on f'_p, and the corresponding final result, f_p, may not be greater or lower than f_o which may appear to lead to a meaningless

order in (5). Nevertheless, as will seen further, there are signal extrema that, in such a case, cannot be removed by σ_p but only by σ_o or some other $\sigma < \sigma_p$. For the moment, we can intuitively consider that for successive simplifications $\#R_{\min}(f_p) \geq \#R_{\min}(f_o)$, where $R_{\min}(f)$ is the set of regional minima in f.

The same considerations can be achieved for $\sigma_p < \sigma_o$, what completes the reasoning for the order relation in (5), although it still lacks a proof for case (v). To further consider cases (i-v), let us enunciate our simplification operator.

5 Controlled Signal Simplification

From the previous discussions, it is clear that the simplification function, Γ, does not have much control over which extrema it suppresses. Nevertheless, given a $t \in S$, as before, it is possible to limit the simplification process solely to those extrema influenced by t. In this regard, the range of the signal f is extended to $\mathbb{R} \cup \{-\infty, \infty\}$ and an operator τ is constructed. When working on regional minima, τ is as follows:

$$\hat{f}(x) = \begin{cases} f(x) & \text{if } x \in R_{\min}(f), \\ \infty & \text{otherwise.} \end{cases} \tag{7}$$

$$\tau(f,t) = \inf\left(\varepsilon\left(\hat{f}, g_t\right)^{l_t}, f\right). \tag{8}$$

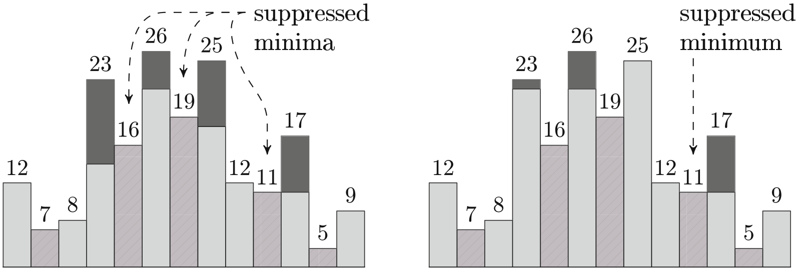
The dual case, when working on regional maxima, is given by:

$$\breve{f}(x) = \begin{cases} f(x) & \text{if } x \in R_{\max}(f), \\ -\infty & \text{otherwise.} \end{cases} \tag{9}$$

$$\tau(f,t) = \sup\left(\delta\left(\breve{f}, g_t\right)^{l_t}, f\right), \tag{10}$$

where l_t indicates the number of erosions/dilations performed with the elementary structuring function g_t described in (6). The τ operator is more restrictive than Γ in the sense that the former considers the entire signal only after the erosions/dilations have already been performed.

Figure 3 illustrates the application of the operators Γ and τ for regional minima. Without loss of generality, we assume the signal is 2-connected, and the function z relates each extremum, e, only to the extrema immediately to the left and to the right of e (save edge cases). The original signal f is the same for both examples. Based on (4), the tuple $t = (6, 1)$ is obtained from the two right-most regional minima (in this case, and according to (6), the structuring function is given by the one-dimensional discrete function $g =$ [-6 0 -6]). Observe how in Fig. 3a the minima at $f(x) = 19$ and $f(x) = 16$ are suppressed by $\Gamma(f,t)$ although they only relate to those extrema that produce $\sigma < 6$. On the other hand, the operator τ provides a stronger condition for the simplification.



(a) Operator $\Gamma(f,t)$ (b) Operator $\tau(f,t)$

Fig. 3. Simplification of a signal f with the tuple $t = (6,1)$ by considering regional minima. In both cases, f is represented by the whole drawing, the lighter area is the final simplified f, and the hatched bars indicate the original regional minima.

As shown in Fig. 3b, the only suppressed minimum by the tuple t is the supremum (point at $f(x) = 11$) of the minima originating this tuple.

Let us now consider how the value l impacts the simplification process, by first approaching the less restrictive operator Γ.

Proposition 1. *If a tuple $t \in S$ is selected, then at most l_t erosions (dilations) with g_t are needed to suppress a minimum (maximum) using $\Gamma(f,t)$.*

Proof. The tuple t was created due to a relation between two regional extrema X and Y. The need for $l_t = l(X,Y)$ erosions/dilations comes from the elementary size of g_t and the propagation of values along l_t vertices in graph G. It remains to prove that the corresponding suppression takes place.

Let $X \leq Y$ and consider the suppression for regional minima. For clarity, let us rewrite $\varepsilon(f,g_t)^{l_t}$ as $\varepsilon(\varepsilon(\ldots),g_t)$, where the *dots* represent $l_t - 1$ successive erosions with g_t. The application of the chain rule $l_t - 1$ times gives

$$\varepsilon(\varepsilon(\ldots),g_t) = \varepsilon\left(f, \delta(g_t,g_t)^{l_t-1}\right),$$

where $\delta(g_t,g_t)^0 = g_t$. Let g'_t be the resulting g after the above composition. From the definition of dilation together with (6), and after g_t being composed $l_t - 1$ times, we have:

$$m_g = |\inf(g'_t)| = \sigma_t \times l_t.$$

Also, $\sigma_t \times l_t = f(b \in Y) - f(a \in X)$ and the neighborhood extension of g'_t has increased from 1 (in the elementary g_t) to l_t. Since m_g holds the difference between the two involved extrema, $m_g + f(a \in X) = f(b \in Y)$.

Finally, we conclude the proof based on the erosion definition. Consider the path P between the above sets X and Y, as indirectly stated in Definition (2). Then:

1. $\forall p \in P \colon \varepsilon(f, g'_t)(p) \leq f(b \in Y)$;
2. By ordering the vertices $p \in P$ between X and Y, based on the number of arcs needed to reach $a \in X$ (the lower minimum), and assuming $\inf(P) \geq f(a)$, then for $i = \{1, \ldots, \#P - 1\}$, $[\Gamma(f,t)](p_i)$ has an upper bound of $\inf\{f(p_i), f(a) + i\sigma_t\}$ and $[\Gamma(f,t)](p_i) \leq \inf\{f(p_{i+1}), f(a) + (i+1)\sigma_t\}$, since $i\sigma_t < (i+1)\sigma_t$. On the other hand, if $\inf(P) \ngeq f(a)$, then less than l_t operations are needed to suppress $f(b \in Y)$. □

As illustrated before, this final result implies the signal simplification through the definition of ramps between minima. This same proof can be done in a complementary way for $X \geq Y$ and by duality for the regional maxima. As shown next, the same aspect regarding the separation l between extrema is equally valid for the more restrictive operator τ.

Proposition 2. *If a tuple $t \in S$ is selected, then at most l_t erosions (dilations) with g_t are needed to suppress a minimum (maximum) using $\tau(f,t)$.*

Proof. The need for l_t operations is in the proof of Proposition 1. As before, we will prove the suppression of extrema for regional minima. Let X and Y, where $X \leq Y$, be the minima originating the tuple t. Let g'_t be the structuring function obtained as in Proposition 1. Again, let P be the path between X and Y, defined as the byproduct of Definition 2. Consider the indices i as in the proof for Proposition 1. Now, assume there is no minima in P related either to $f(a \in X)$ or $f(b \in X)$ – otherwise, this proof should be applied piecewise for each segment in P violating this assumption.

From (7), $\hat{f}(p) = \infty$, $\forall p \in P$. Thus, a single erosion with g'_t gives:

1. $\forall p \in P \colon \varepsilon(\hat{f}, g'_t)(p) \leq f(b \in Y)$
2. $\varepsilon(\hat{f}, g'_t)(p_i) \leq \varepsilon(\hat{f}, g'_t)(p_{i+1})$

At this point, we have to show that the operator in (8) maintains the suppression of minima. To this end, it is needed to consider two specific configurations for the path between X and Y, namely:

a) The path P is monotonically decreasing from $p_1 \in P$ to $b \in Y$. Hence, $\varepsilon(\hat{f}, g'_t)(p) \leq f(p)$, $\forall p \in P$;

b) The path P is monotonically increasing from $a \in X$ to $p_i \in P$ and then monotonically decreasing from $p_{i+1} \in P$ to $b \in Y$. Suppose $f(p_{i+1}) = \sup\{f(p) \mid p \in P\}$. But $f(p_{i+1}), \ldots, f(b \in Y)$ must be higher than $f(a)$, otherwise $X \nleq Y$. Therefore, case (a) also holds for the monotonically decreasing part here. Besides, $f(p_{i+1})$ is transformed to $f(a) + k\sigma_t$, where k is the number of arcs to the minimum X, and is the new smallest value for this part. Now, to guarantee a non-decreasing path from X to Y, it is needed to show that $[\tau(f,t)](p_i) < [\tau(f,t)](p_{i+1})$. Since $[\tau(f,t)](p_i) = \inf(\varepsilon(\hat{f}, g'_t)(p_i), f(p_i))$ $= \inf(f(a) + r\sigma_t, f(p_i))$, and $r < k$, then $[\tau(f,t)](p_i) < [\tau(f,t)](p_{i+1})$. □

The same above considerations for $X \geq Y$ and concerning duality are applied here. Finally, the next proposition explicits the extrema affected by the operator τ.

An extension of this proposition, relating the evolution of the regional extrema along the simplification step, yields the result discussed in Sect. 4, item (v). For a sake of space, a proof is not presented here, but most of it can be given by observing the support of the extended g_t (as in earlier proofs) and considering the progressive activity discussed in Sect. 4.

Proposition 3. *A simplification $\tau(f, o)$, where $o \in S$ and $\sigma_o \neq 0$, suppresses only the minima/maxima related with the tuple o and also with a tuple $p \in S$ so that $p \prec o$ and $l_o \geq l_p$.*

Observe that the order relation imposed on S turns the selection of a tuple $t \in S$ in a kind of meaningful threshold (but note that this information alone is not enough to properly indicate which tuple(s) should be used in a given application). The minimal element in S suppresses the influenced extrema of maximum divergence, and the further we go from there, extrema of lesser disparity are suppressed. Also note that in the uttermost case, there may be relations among the regional extrema of equal height, which reduces g_t to a planar SF, according to (4) and (6). In this specific situation, the flat zones of a signal are extended.

6 Examples

This section illustrates the application of the τ operator in two instances. In both of them, a regular 4-connected grid graph is considered and the z function is given by the Meyer's watershed transform mentioned in Sect. 3.

The following example highlights some of the first regions affected by the different simplification levels. In natural and non-predominantly noisy images, it is common to have fuzzy borders between objects and background defined by regions represented by significant peaks and valleys. In such a case, and based on (4), we expect to obtain high values of σ together with small distances l, around this uncertain area. According to the relation $\mathcal{R}$, these tuples are among the first in the considered order, so that a simplification at almost any level is bound to affect the related extrema and their corresponding connecting paths giving origin to such tuples. As can be seen in Fig. 4, the simplification progressively affects the image regions, where initially most of the modifications happens close to their edges.

Figure 5 shows an example of image simplification which can be considered in segmentation tasks. The problem concerns the segmentation of white blood cells nuclei and illustrates how the proposed method can be applied to define good marker images, commonly used in some morphological transformations.

The input image is particularly noisy, the gradient around the nuclei is weak, and there are small dark regions unrelated to the cells. To partially handle the noise and and also to remove part of the dark blobs, we initially considered the regional maxima in the simplification process which defined the strongly simplified image in Fig.5b. This simplified version is then used as a marker to dually reconstruct the original image, yielding a more homogeneous background where most of the noise is filtered out, as shown in Fig. 5c. Now, to isolate the dark

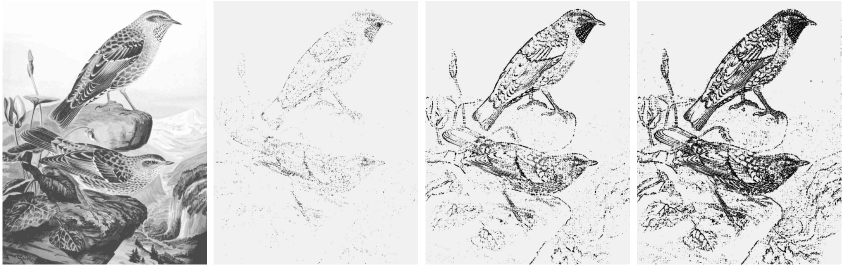


Fig. 4. Primary regions affected by the method. The leftmost image is the original one, the others represent the difference between the original and the simplified image using minima with increasing t_i ordered by $\mathcal{R}$, i = $100th$, $500th$, $1000th$.

regions corresponding to the nuclei, we progressively simplify, till idempotence, the regional minima of the simplified signal by considering the minimal element in S at each iteration (Fig. 5d). Again, by using this simplified image as a marker to reconstruct the signal in Fig. 5c, a useful image is obtained (Fig.5e) and it can be applied, for example, in a segmentation procedure. The final result in Fig.5f is a typical morphological segmentation by watershed in which the regional minima in Fig.5e are imposed to its gradient image [2].

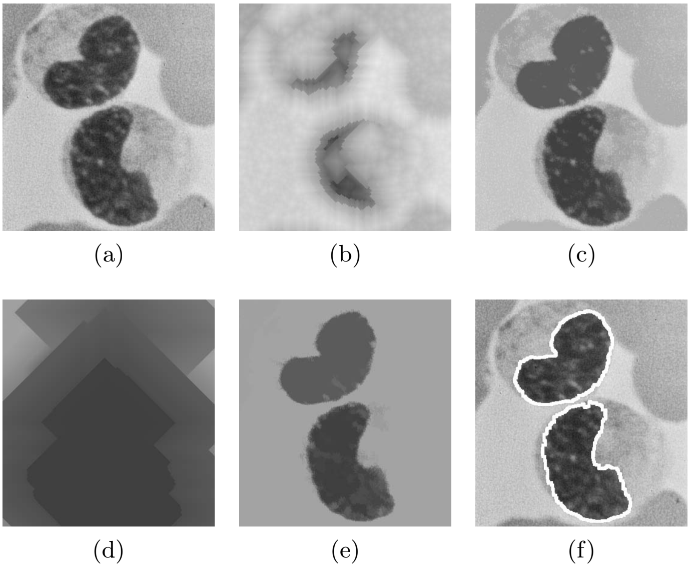


Fig. 5. White blood cell nucleus segmentation. (a) Original image f. (b) Strongly simplified image after 50 simplification steps of the regional maxima with the $100th$ tuple. (c) Dual reconstruction of (a) from (b). (d) Simplification of the regional minima in (c), till idempotence, with the minimal element in S at each iteration. (e) Reconstruction of (c) from (d). (f) Final segmentation of the nuclei by well-known morphological tools.

7 Conclusion

A strict total order relation $\mathcal{R}$ based both on the height difference and separation of the extrema in signal was defined in this paper. This relation can be guided by a nominated function which decides whether to relate two extrema or not. Also, $\mathcal{R}$ settles a non-decreasing activity with respect to the extrema, allowing for a better controlled simplification. From this resulting relation, a concave structuring function g_t, which is easily adaptable to a general graph, is constructed. With g_t, the defined operator τ simplifies the input by means of a well-behaved extrema suppression. Some short examples give a preliminary idea of how the simplification approach can be used in real applications.

Current work on this subject concerns the formalization of a scale-space based on the simplification defined by τ. Also, despite the fact that the operators τ and Γ work on graphs, the above mentioned examples considered each pixel in the digital image as a vertex. Naturally, it remains to investigate how these operators behave in more varied and complex configurations. Other shapes of structuring functions might be of interest too, as well as the use of the relation $\mathcal{R}$ with other functions besides erosion and dilation.

Acknowledgements. The authors thank to Brazilian agencies CNPq and FAPESP for financial support.

References

1. Bertrand, G.: On the Dynamics. Image Vision Comput. 25(4), 447–454 (2007)
2. Beucher, S., Meyer, F.: The Morphological Approach to Segmentation: The Watershed Transformation. In: Mathematical Morphology in Image Processing. Marcel Dekker (1993)
3. van den Boomgaard, R., Dorst, L., Makram-Ebeid, S., Schavemaker, J.G.M.: Quadratic Structuring Functions in Mathematical Morphology. In: Mathematical Morphology and its Applications to Image and Signal Processing. Kluwer (1996)
4. Dorini, L.B., Leite, N.J.: Multiscale Morphological Image Simplification. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 413–420. Springer, Heidelberg (2008)
5. Grimaud, M.: A New Measure of Contrast: The Dynamics. In: Proc. SPIE, Image Algebra and Morphological Image Processing III, vol. 1769, pp. 292–305 (1992)
6. Hagyard, D., Razaz, M., Atkin, P.: Analysis of Watershed Algorithms for Greyscale Images. ICIP 3, 41–44 (1996)
7. Jackway, P.T., Deriche, M.: Scale-Space Properties of the Multiscale Morphological Dilation-Erosion. IEEE Trans. Pattern Anal. Mach. Intell. 18(1), 38–51 (1996)
8. Lindeberg, T., ter Haar Romeny, B.M.: Linear Scale-Space: Basic Theory. In: Geometry-Driven Diffusion in Computer Vision. Kluwer (1994)
9. Meyer, F.: Levelings, Image Simplification Filter for Segmentation. J. Math. Imaging Vision, 59–72 (2004)
10. Meyer, F., Serra, J.: Contrasts and Activity Lattice. Signal Process 16(4), 303–317 (1989)

11. Pitas, I., Venetsanopoulos, A.N.: Order Statistics in Digital Image Processing. P. IEEE 80(12), 1893–1921 (1992)
12. Serra, J.: Image Analysis and Mathematical Morphology, vol. 1. Academic Press (1982)
13. Serra, J., Salembier, P.: Connected Operators and Pyramids. In: Proc. SPIE, Image Algebra and Morphological Image Processing IV, vol. 2030, pp. 65–76 (1993)
14. Silva, A.G., Lotufo, R.A.: Efficient Computation of New Extinction Values from Extended Component Tree. Pattern Recogn. Lett. 32(1), 79–90 (2011)
15. Vachier, C., Vincent, L.: Valuation of Image Extrema Using Alternating Filters by Reconstruction. In: Neural, Morphological, and Stochastic Methods in Image and Signal Processing, pp. 94–103 (1995)
16. Wilson, S.S.: Vector Morphology and Iconic Neural Networks. IEEE Trans. Syst. Man Cybern. 19(6), 1636–1644 (1989)

Two Applications of Shape-Based Morphology: Blood Vessels Segmentation and a Generalization of Constrained Connectivity

Yongchao Xu[1,2], Thierry Géraud[1], and Laurent Najman[2]

[1] EPITA Research and Development Laboratory (LRDE), France
{yongchao.xu,thierry.geraud}@lrde.epita.fr
[2] Université Paris-Est, LIGM, Équipe A3SI, ESIEE, France
l.najman@esiee.fr

Abstract. Connected filtering is a popular strategy that relies on tree-based image representations: for example, one can compute an attribute on each node of the tree and keep only the nodes for which the attribute is sufficiently strong. This operation can be seen as a thresholding of the tree, seen as a graph whose nodes are weighted by the attribute. Rather than being satisfied with a mere thresholding, we propose to expand on this idea, and to apply connected filters on this latest graph. Consequently, the filtering is done not in the space of the image, but on the space of shapes built from the image. Such a processing, that we called *shape-based morphology* [30], is a generalization of the existing tree-based connected operators. In this paper, two different applications are studied: in the first one, we apply our framework to blood vessels segmentation in retinal images. In the second one, we propose an extension of constrained connectivity. In both cases, quantitative evaluations demonstrate that shape-based filtering, a mere filtering step that we compare to more evolved processings, achieves state-of-the-art results.

1 Introduction

Mathematical morphology, as originally developed by Matheron and Serra [23], proposes a set of morphological operators based on structuring elements. Later, Salembier and Serra [21], followed by Breen and Jones [3], proposed morphological operators based on attributes, rather than on elements. Such operators rely on transforming the image into an equivalent representation, generally a tree of components (*e.g.*, level sets) of the image; such trees are equivalent to the original image in the sense that the image can be reconstructed from its associated tree. Filtering then involves the design of a shape attribute that weights how much a node of the tree fits a given shape. Two different approaches for filtering the tree (and hence the image) have been proposed: the more evolved approach consists in pruning the tree by removing whole branches of the tree, and is easy to apply if the attribute is increasing on the tree (*i.e.*, if the attribute is always stronger for the ancestors of a node). This process is depicted in the black path in Fig. 1.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 390–401, 2013.

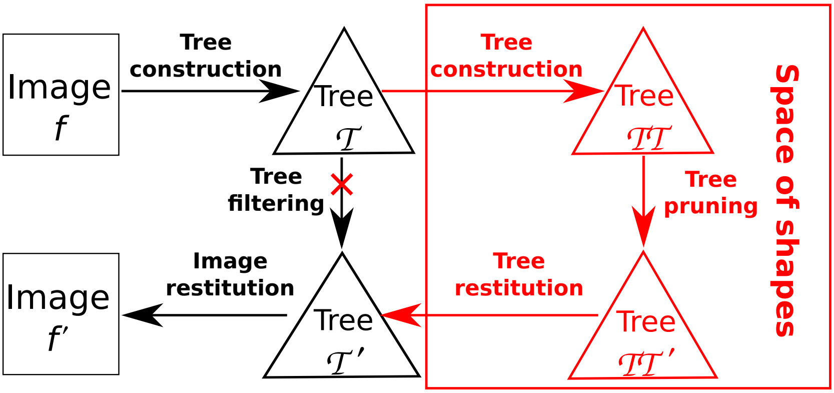
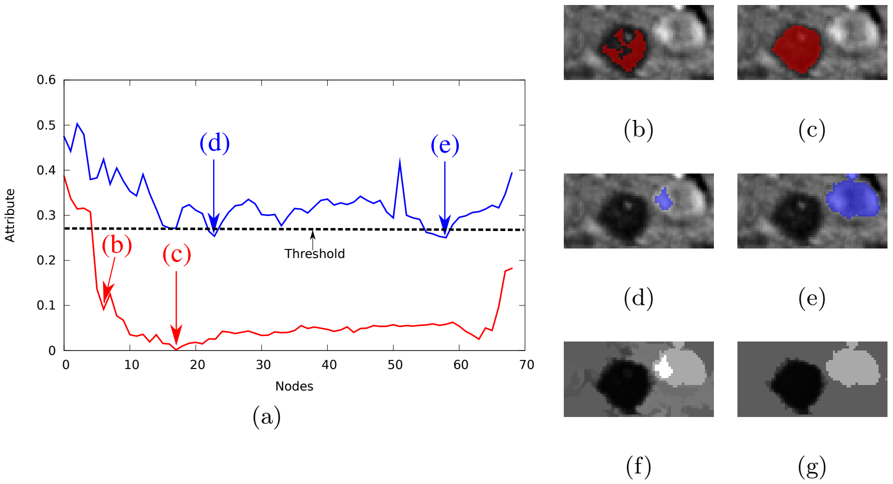


Fig. 1. Classical connected operators (black path) and our proposed shape-based morphology (black+red path)

However, most shape attributes are not increasing. When the attribute is not increasing, three strategies have been proposed (min, max, Viterbi; see [22] for more details). They all choose a particular node on which to take the decision, and remove the whole subtree rooted at this node. While it may give interesting results in some cases, it does not take into account the possibility that several relevant objects can have some inclusion relationship, meaning that they are on the same branch of the tree (*e.g.*, a ring object in a tree of shapes, see Fig. 3.a).

In the simplest approach, one simply removes the nodes of the tree for which the attribute is lower than a given threshold [27]. Such a thresholding does not take into account the intrinsic parenthood relationship of the tree. Moreover it is often impossible to retrieve all expected objects with one unique threshold. Fig. 2 shows the evolution of a shape attribute, the circularity, along two branches of the tree of shapes [15]. The light round shape and the dark one are both meaningful round objects compared to their context. However, their attribute values are very different. In order to obtain the light one, a higher threshold is applied, but some non-desired shapes appear, the ones in the background in Fig. 2.f.

The founding idea of shape-based morphology is to apply connected filters on the space of all the components of the image, such space being structured into a graph by the parenthood relationship (i.e., the neighbors of a node are its children and its parent). This process is illustrated by the black+red path of Fig. 1. This surprising and simple idea has several deep consequences that were first exposed in [30], where it is shown that this framework encompasses the usual attribute filtering operators [30]. Novel connected filters based on non-increasing criterion can also be proposed. When the first tree $\mathcal{T}$ is respectively a Min-tree or a Max-tree, such filters are new morphological lower or upper levelings. When the first tree $\mathcal{T}$ is the tree of shapes, we introduce a novel family of self-dual connected filters that we call *morphological shapings*. The proposed framework can also be used to produce extinction-based [28] saliency maps [18,16].



(a)

(b) (c)

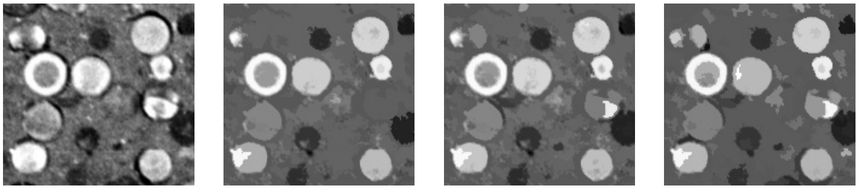
(d) (e)

(f) (g)

Fig. 2. (a) Evolution of "circularity" on two branches of $\mathcal{T}$; (b to e): Some shapes; (f) Attribute thresholding; (g) Shaping

In general, shape-based morphological operators provide better results than threshold-based approaches. As stated above, when we want to process both upper and lower level sets at the same time, we use as $\mathcal{T}$ the tree of shapes, and the such created operator is called a morphological shaping. In both Fig. 2 and Fig. 3, the attribute $\mathcal{A}$ is the circularity. The result of the shaping on Fig. 2.a, shown in Fig. 2.g, looks indeed better than the one of Fig. 2.f. In Fig. 3, we compare this extinction-based self-dual shaping approach with a variant of the state-of-the-art thresholding approach [27]. When the threshold of $\mathcal{A}$ is low, some objects do not appear (Fig. 3.b). To get all the expected objects, we have to set a high threshold; however, in this case, too many unwanted objects are present (Fig. 3.c). With our shaping filter, all the expected objects can be found, as depicted in Fig. 3.d.

In this paper, we propose to detail how the framework can be used for two different types of applications. The rest of this paper is organized as follows. An application of our proposed shape-based upper levelings to blood vessels



(a) Input image. (b) Low threshold. (c) Higher threshold. (d) Our shaping.

Fig. 3. Comparison of extinction-based shapings with attribute thresholding

segmentation in retinal images is explained in Section 2. In Section 3, we detail how to extend the constraint connectivity framework first introduced in [24]. Finally we discuss and conclude in Section 4.

2 Blood Vessels Segmentation in Retinal Images

Blood vessels segmentation is a very important task in retinal images analysis. Unlike classical linear filters, connected operators perfectly preserve the location and the shape of the contours, which provides a motivation for using them.

Many existing methods work on the green channel of the color retinal image. To improve the visibility of the blood vessels, for each color retinal image f_c, a black top-hat transform is applied to the green channel f_g. When a mask of eye fundus is available, we combine it with the black top-hat f_t. We thus obtain an image f_i in which the blood vessels are visible: indeed, the main structures of the blood vessels are present in the Max-tree $\mathcal{T}$ of f_i (the connected components of the upper level sets of f_i, [22]).

For each connected component represented by some node $\mathcal{N}_k$ of the Max-tree $\mathcal{T}$, we compute a shape attribute $\mathcal{A}$ characterizing the blood vessels, which are usually long and thin structures. The attribute used here is the elongation $\mathcal{A}_e$:

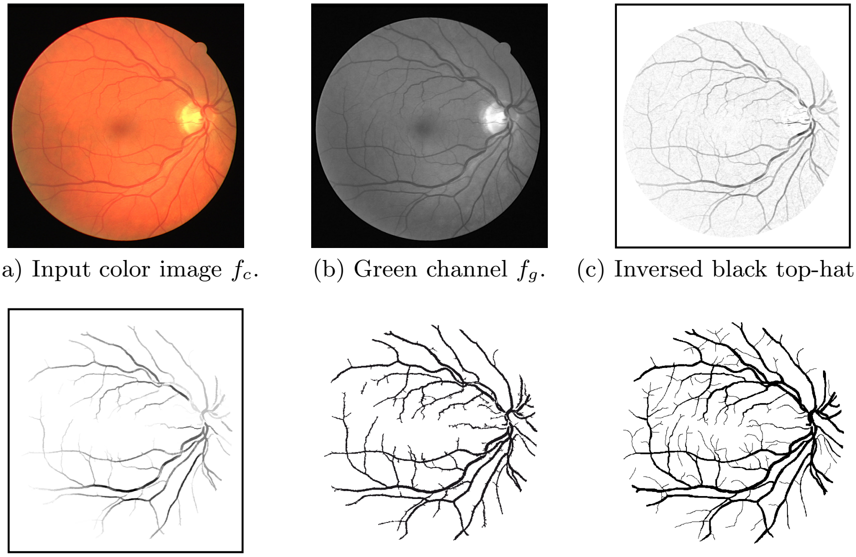
$$\mathcal{A}_e(\mathcal{N}_k) = |\mathcal{N}_k|/(\pi \times l_{max}^2), \tag{1}$$

where $|.|$ denotes the cardinality and l_{max} denotes the length of the largest axis of the best fitting ellipse for the connected component represented by $\mathcal{N}_k$. Since the blood vessels are long and thin, nodes having a low attribute $\mathcal{A}_e$ correspond to the blood vessels.

The core of this application is the filtering of the Max-tree $\mathcal{T}$. A mere thresholding of the elongation $\mathcal{A}_e$ is not sufficient, often giving unwanted objects (noise). However, a very low thresholding value t_{min} on $\mathcal{A}_e$ ensures that thresholded nodes are blood vessels. Those initial extracted nodes are used as seeds in the sequel. We then apply a morphological filtering with a depth criterion: using the Min-tree $\mathcal{TT}$ of the node-weighted graph $(\mathcal{T}, \mathcal{A}_e)$, we only preserve the nodes that have a certain depth d_0 in $\mathcal{TT}$ and that furthermore contain the seeds. The connected components represented by the preserved nodes are considered as the segmented blood vessels. The whole process is one of the many variants of shape-based upper levelings [30].

An example of this blood vessels segmentation methods is given in Fig. 4. As compared with the manually segmented blood vessels segmentation (Fig. 4.f), the elongation-based upper leveling (Fig. 4.e) correctly extracts most of the blood vessels.

We have tested this specific shape-based upper leveling on the Digital Retinal Images for Vessel Extraction (DRIVE) database [5], [25] and on the STructured Analysis of the Retina (STARE) database [26], [8]. DRIVE is a database assembled in the Netherlands from a diabetic retinopathy screening program.



(a) Input color image f_c. (b) Green channel f_g. (c) Inversed black top-hat f_i.

(d) Inversed filtering result. (e) Segmented blood vessels. (f) Manual segmentation.

Fig. 4. Illustration of the complete process of blood vessels segmentation in retinal image using elongation-based upper leveling (a-e); (f): Manual segmentation

It includes 40 color fundus images of 584 × 565 pixels, captured using a 45° field-of-view fundus camera. The 40 color images are divided into a training set and a test set, both containing 20 images. For the training images, a single manual segmentation of the vasculature is available. For the test cases, two manual segmentations are available: one is used as gold standard; the other one can be used to compare computer generated segmentations with those of an independent human observer. The complete database contains seven pathological cases (four in the test set and three in the training set). The STARE database contains 20 images captured using a TopCon TRV-50 fundus camera at 35° FOV, and digitized to 700 × 605 pixels, 8 bits per RGB channel. A manual segmentation is available for each image of the database. Masks of the eye fundus, derived from the matched spatial filter [8], are also available. Note that, among the 20 images, 10 images are abnormal.

Fig. 5 and 6 show four segmentation results respectively from the DRIVE database and the STARE database. Qualitatively, most of the blood vessels are correctly extracted, although some noise points at the end of the vessels are also kept, and some very thin blood vessels are missed.

Quantitative assesment is based on three performance measurements named respectively *sensitivity*, *specificity* and *accuracy* [25]. Sensitivity measures the true positive rate (TPR), specificity measures the true negative rate (TNR), and accuracy measures the rate of pixels correctly classified. These measurements are

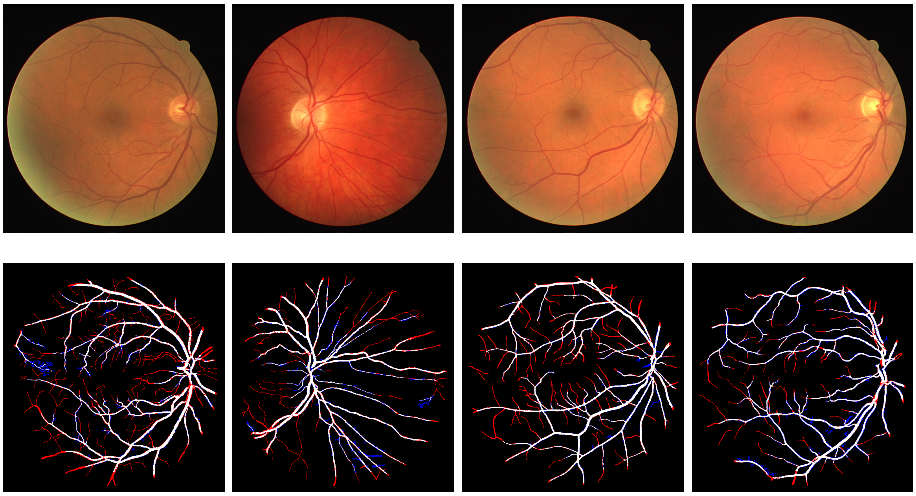


Fig. 5. Illustration of four segmented blood vessels from the 20 test retinal images of the DRIVE database. Top: Input color retinal images; Bottom: Corresponding segmented results. (White pixels: true positive; Black pixels: true negative; Blue pixels: false positive; Red pixels: false negative.)

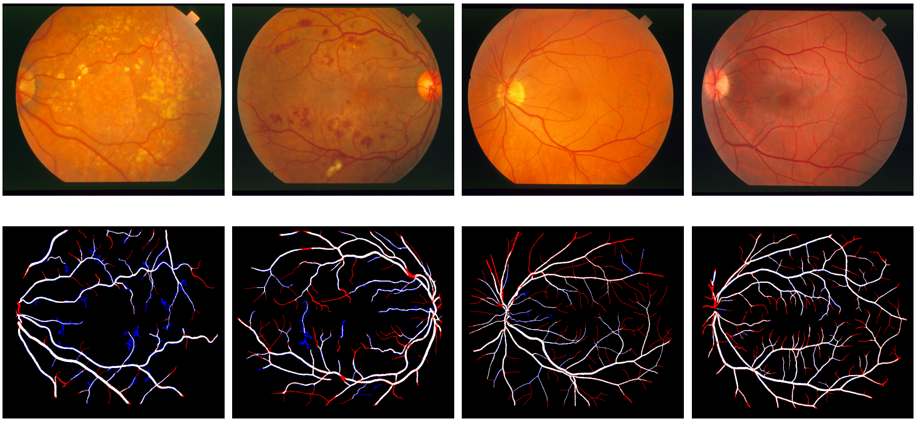


Fig. 6. Illustration of four segmented blood vessels from the 20 test retinal images of the STARE database. Top: Input color retinal images; Bottom: Corresponding segmented results. (White pixels: true positive; Black pixels: true negative; Blue pixels: false positive; Red pixels: false negative.)

Table 1. Benchmark of different blood vessels segmentation approaches on DRIVE and the STARE database

	DRIVE			STARE		
Method	TPR	TNR	Accuracy	TPR	TNR	Accuracy
2$^{\text{nd}}$ Expert	0.7761	0.9725	0.9473 (0.0048)	0.8949	0.9390	0.9354 (0.0171)
mendonça [14]	**0.7344**	**0.9764**	**0.9452 (0.0062)**	0.6996	0.9730	0.9440 (0.0142)
Staal [25]	0.7193	0.9773	0.9441 (0.0057)	-	-	-
Niemeijer [19]	0.6793	0.9801	0.9416 (0.0065)	-	-	-
Our	0.6924	0.9779	0.9413 (0.0078)	0.7149	0.9749	0.9471 (0.0114)
Zana [31]	0.6696	0.9769	0.9377 (0.0078)	-	-	-
Al-Diri [1] -	-	-	0.9258 (0.0126)	-	-	-
Jiang [9]	0.6478	0.9625	0.9222 (0.0070)	**-**	**-**	**0.9513**
Perez [12]	0.7086	0.9496	0.9181 (0.0240)	-	-	-
Hoover [8]	-	-	-	0.6751	0.9567	0.9267 (0.0099)

defined below as:

$$sensitivity = TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{2}$$

$$specificity = TNR = \frac{TN}{N} = \frac{TN}{TN + FP} \tag{3}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4}$$

where TP stands for true positive, FP for false positive, TN for true negative and FN for false negative.

A benchmark of different approaches (including ours) is provided in Table 1. It shows the good performance of our proposed elongation-based upper leveling for both databases. In the case of the DRIVE database, our result is slightly under the best results given by the method of Mendonça [14]. Note also that the approaches of Staal [25] and Niemeijer [19] are supervised approaches. On the STARE database, our proposed method performs also very well, and is even better than the method of Mendonça [14]. Both methods give results that are very close to one of the second human observer. Table 2 shows that our method is more robust than others, in the sense that it performs equivalently on both abnormal and normal images.

Last, note that the proposed elongation-based upper leveling is only a "simple" filtering step, whereas other approaches are more complicated. Besides, our process is not complete, since further post-processing can improve the results.

3 Extending Constrained Connectivity

From an algorithmic point of view, constrained connectivity [24] is the application of an increasing criterion (*e.g.*, the range) on the Min-tree $\mathcal{T}$ of the minimum spanning tree (MST) [16,17]. In the litterature, this tree $\mathcal{T}$ has been called the

Table 2. Benchmark of vessels segmentation methods (STARE images - Normal versus abnormal cases)

Method	Sensitivity	Specificity	Accuracy
Normal cases			
2^{nd} human observer	0.9646	0.9236	0.9283 (0.0100)
Shape-based upper leveling	**0.7178**	**0.9802**	**0.9493 (0.117)**
mendonça [14]	0.7258	0.9791	0.9492 (0.0122)
Hoover [8]	0.6766	0.9662	0.9324 (0.0072)
Abnormal cases			
2^{nd} human observer	0.0.8252	0.9544	0.9425
Shape-based upper leveling	**0.7120**	**0.9696**	**0.9447 (0.0106)**
mendonça [14]	0.6733	0.9669	0.9388 (0.0150)
Hoover [8]	0.6736	0.9472	0.9211 (0.0091)

α-tree [20]. In this section, we apply our framework with a non-increasing criterion $\mathcal{A}$ derived from a popular work [6]. Precisely, we use for $\mathcal{T}$ a binary Min-tree of the MST [17]. The dissimilarity used for the MST is the maximal distance of the red, blue, and green channels taken independently.

The attribute $\mathcal{A}$ is derived from [6], where the authors propose a region merging process that follows the edges of the MST by increasing order of the weights (dissimilarity). When an edge $\{x, y\}$ is considered, they search for the regions X and Y that respectively contain the points x and y. The regions X and Y are merged if

$$\text{Diff}(X, Y) \; < \; \min\{\, \text{Int}(X) + \frac{k}{|X|}, \, \text{Int}(Y) + \frac{k}{|Y|}\}, \tag{5}$$

where $|.|$ denotes the cardinality, $\text{Diff}(X, Y)$ is the minimum weight of the edge connecting the two regions X and Y, $\text{Int}(X)$ is the largest weight in the MST of the region X, and k is a parameter favouring the merging of small regions (a large k causes a preference for larger components). However, k is not a scale parameter in the sense of the *causality principle*: as shown in [7] a contour present at a scale k_1 is not always present at a scale $k_2 < k_1$. The merging criterion defined by Eq. (5) depends on the parameter k at which the regions X and Y are observed. So let us consider the attribute $\mathcal{A}$ as the k defined by $k = \max\Big\{\big(\text{Diff}(X, Y) - \text{Int}(X)\big) \times |X|, \big(\text{Diff}(X, Y) - \text{Int}(Y)\big) \times |Y|\Big\}$. That is to say, for each node $\mathcal{N}_k$, let $\mathcal{N}_k^{c1}$ and $\mathcal{N}_k^{c2}$ be the two children of $\mathcal{N}_k$ in $\mathcal{T}$, then the attribute $\mathcal{A}$ for node $\mathcal{N}_k$ is given by

$$\begin{aligned}\mathcal{A}(\mathcal{N}_k) = \max\Big\{ \; & \big(\text{Diff}(\mathcal{N}_k^{c1}, \mathcal{N}_k^{c2}) - \text{Int}(\mathcal{N}_k^{c1})\big) \times |\mathcal{N}_k^{c1}|, \\ & \big(\text{Diff}(\mathcal{N}_k^{c1}, \mathcal{N}_k^{c2}) - \text{Int}(\mathcal{N}_k^{c2})\big) \times |\mathcal{N}_k^{c2}| \Big\}. \end{aligned} \tag{6}$$

The maxima of the attribute $\mathcal{A}$ correspond to meaningful regions. We thus compute a Max-tree $\mathcal{TT}$ on the node-weigthed graph $(\mathcal{T}, \mathcal{A})$, with which we can filter this graph. The contours of the flat zones of one level of filtering of $(\mathcal{TT}, \mathcal{A})$ provides a segmentation of the original image. Computing all levels of filtering

Table 3. Preliminary region benchmarks on the BSDS500

Method	GT Covering			Prob. Rand. Index	
	ODS	OIS	Best	ODS	OIS
FH [6]	0.43	0.53	0.68	0.76	0.79
Guimarães [7]	0.46	0.53	0.60	0.76	0.81
Ours	**0.50**	**0.57**	**0.66**	**0.77**	**0.82**

produces a hierarchy of segmentations. A good representation of such a hierarchy is a saliency map [18,16], which is equal to the sum of all the contours of every filtered image. An efficient algorithm for computing the saliency map in our framework, based on the notion of extinction value [28], will be provided in an extend version of this paper. The saliency map defines a duality between closed, non-self intersecting weighted contours and a hierarchy of regions. Low levels of the hierarchy correspond to weak contours, thus to over-segmentations. High levels of the hierarchy correspond to strong contours, hence to under-segmentations. Moving between levels gives a continuous trade-off between those two extremes. A given level can be seen as an *observation scale* at which we consider the image.

We have tested our extended constrained connectivity framework on the Berkeley Segmentation Dataset BSDS500 [2], an extension of the BSDS300 [13]. The dataset consists of 500 natural images divided into 200 test images, 200 images for training, and 100 validation images, together with human annotations.

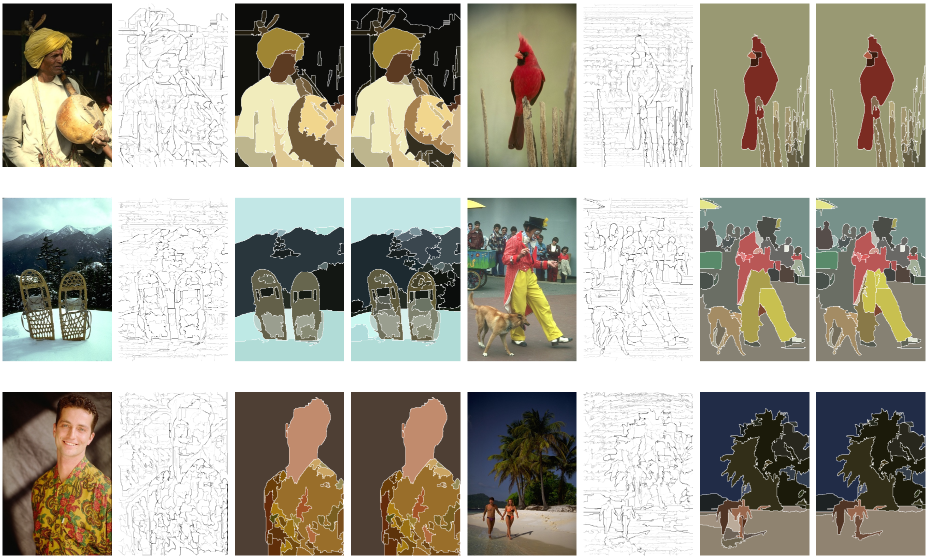


Fig. 7. Hierarchical segmentation results on the BSDS500. From left to right: Input image, saliency map, and segmentations at the optimal dataset scale (ODS) and at the optimal image scale (OIS).

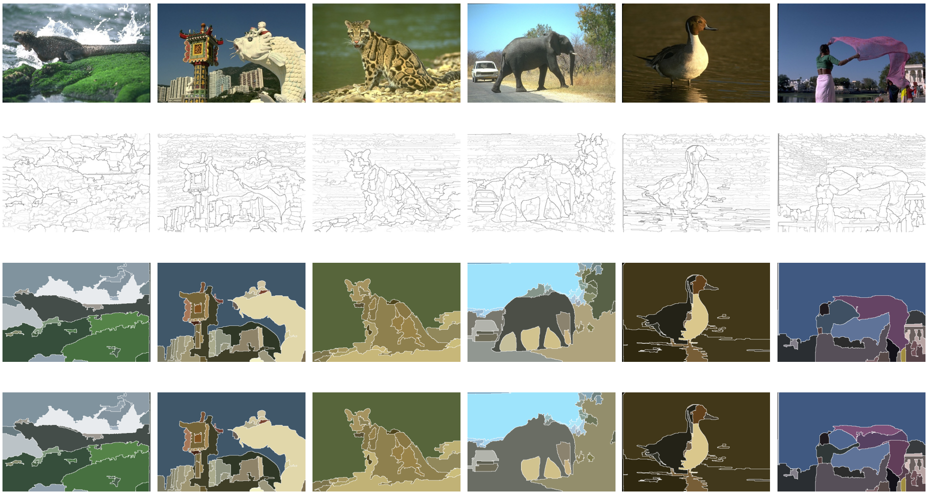


Fig. 8. Additional hierarchical segmentation results on the BSDS500. From top to bottom: Input mage, saliency map, and segmentations at the optimal dataset scale (ODS) and at the optimal image scale (OIS).

Each image is segmented by an average of five different subjects. Fig. 7 and 8 show some saliency maps computed on some images from the BSDS500 dataset. Two evaluation schemes are also provided by the authors. In the first one, the same fixed threshold level (observation scale) is used for all saliency maps in the dataset; we refer to it as the *optimal dataset scale (ODS)*. In the second one, we evaluate the performance using an image-dependent threshold for each saliency map; we refer to this choice as the *optimal image scale (OIS)*.

Quantitative evaluation is performed using the region-based performance measurements described in [2], in terms of Ground-Truth Covering criterion and Probabilistic Rand Index. Here, we compare our results with the graph-based image segmentation (Felz-Hutt) [6], and with another method named hierarchical graph based image segmentation (Guimarães *et al.*) [7], also relying on the same criterion popularized by [6]. The comparison is given in Table 3. Our method ranks first, for both the optimal dataset scale (ODS) and for optimal image scale (OIS).

4 Discussion and Conclusion

This paper has presented two applications of shape-based morphology, a generalization of existing tree-based connected operators. The first application uses one of the many shape-based upper levelings. Althought such a filter is but a "simple" filtering step, it gives results almost as good as the second human observer in the case of blood vessels segmentation. The second application is an extension of the constrainted connectivity framework to non-increasing constraints. A quantitative evaluation based on a criterion given in [6] shows that our approach compares favorably to previous works.

The potential of shape-based morphology is tremendous. In this paper, we have hardly skimmed the surface of what this theory has to offer to the scientific community. In [29], we have also used this framework to achieve object segmentation on the tree of shapes. The key idea was to propose an efficient context-based energy estimator whose minima correspond to meaningful objects. Some other criterions that can be used are, for example an energy derived from the number of false alarms [4] or some snake energy [10]. A practical problem is that many minima of such energies do not correspond to meaningful components of the input image. In [29], a morphological closing in the space of shapes helps to filter those spurious minima.

Implementations of shape-based filters are easy thanks to the open-source Milena library [11]. A demonstration is available online from `http://olena.lrde.epita.fr/ICPR2012` (see also `http://olena.lrde.epita.fr/ICIP2012`). More applications will be studied in a forthcoming paper. Properties of those filters, such as conditions for idempotence, will be also studied.

Acknowledgements. This work received funding from the Agence Nationale de la Recherche, contract ANR-2010-BLAN-0205-03 and through "Programme d'Investissements d'Avenir" (LabEx BEZOUT n°ANR-10-LABX-58).

References

1. Al-Diri, B., Steel, D.: An active contour model for segmenting and measuring retinal vessels. IEEE Transactions on Medical Imaging 28(9), 1488–1497 (2009)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. PAMI 33(5), 898–916 (2011)
3. Breen, E., Jones, R.: Attribute openings, thinnings, and granulometries. CVIU 64(3), 377–389 (1996)
4. Cao, F., Musé, P., Sur, F.: Extracting meaningful curves from images. JMIV 22, 159–181 (2005)
5. DRIVE: Digital Retinal Images for Vessel Extraction, `http://www.isi.uu.nl/Research/Databases/DRIVE/`
6. Felzenszwalb, P., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. Comput. Vision 59(2), 167–181 (2004)
7. Guimarães, S.J.F., Cousty, J., Kenmochi, Y., Najman, L.: A hierarchical image segmentation algorithm based on an observation scale. In: Gimel'farb, G., Hancock, E., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., Yamada, K. (eds.) SSPR & SPR 2012. LNCS, vol. 7626, pp. 116–125. Springer, Heidelberg (2012)
8. Hoover, A., Kouznetsova, V., Goldbaum, M.H.: Locating blood vessels in retinal images by piece-wise threshold probing of a matched filter response. IEEE Transactions on Medical Imaging 19, 203–210 (2000)
9. Jiang, X., Mojon, D.: Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. PAMI 25(1), 131–137 (2003)
10. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. IJCV 1, 321–331 (1987)
11. Levillain, R., Géraud, T., Najman, L.: Why and how to design a generic and efficient image processing framework: The case of the Milena library. In: Proc. of ICIP, pp. 1941–1944 (2010), `http://olena.lrde.epita.fr`

12. Martínez-Pérez, M.E., Hughes, A.D., Stanton, A.V., Thom, S.A., Bharath, A.A., Parker, K.H.: Retinal blood vessel segmentation by means of scale-space analysis and region growing. In: Taylor, C., Colchester, A. (eds.) MICCAI 1999. LNCS, vol. 1679, pp. 90–97. Springer, Heidelberg (1999)
13. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision, vol. 2, pp. 416–423 (July 2001)
14. Mendonça, A.M., Campilho, A.: Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. IEEE Trans. Med. Imaging 25(9), 1200–1213 (2006)
15. Monasse, P., Guichard, F.: Fast computation of a contrast-invariant image representation. IEEE Trans. on Image Processing 9(5), 860–872 (2000)
16. Najman, L.: On the equivalence between hierarchical segmentations and ultrametric watersheds. Journal of Mathematical Imaging and Vision 40, 231–247 (2011)
17. Najman, L., Cousty, J., Perret, B.: Playing with kruskal: algorithms for morphological trees in edge-weighted graphs. In: Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.) ISMM 2013. LNCS, vol. 7883, pp. 135–146. Springer, Heidelberg (2013)
18. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. PAMI 18(12), 1163–1173 (1996)
19. Niemeijer, M., Staal, J.J., van Ginneken, B., Loog, M., Abramoff, M.D.: Comparative study of retinal vessel segmentation methods on a new publicly available database. In: Fitzpatrick, J.M., Sonka, M. (eds.) SPIE Medical Imaging, vol. 5370, pp. 648–656. SPIE (2004)
20. 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)
21. Salembier, P., Serra, J.: Flat zones filtering, connected operators and filters by reconstruction. IEEE Trans. on Image Processing 3(8), 1153–1160 (1995)
22. Salembier, P., Wilkinson, M.: Connected operators. IEEE Signal Processing Mag. 26(6), 136–157 (2009)
23. Serra, J.: Image Analysis and Mathematical Morphology, vol. 1. Academic Press, New York (1982)
24. Soille, P.: Constrained connectivity for hierarchical image decomposition and simplification. PAMI 30(7), 1132–1145 (2008)
25. Staal, J.J., Abramoff, M.D., Niemeijer, M., Viergever, M.A., van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging 23(4), 501–509 (2004)
26. STARE: STructured Analysis of the Retina, `http://www.ces.clemson.edu/~ahoover/stare/`
27. 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. PAMI 29(2), 272–285 (2007)
28. Vachier, C., Meyer, F.: Extinction values: A new measurement of persistence. In: IEEE Workshop on Non Linear Signal/Image Processing, pp. 254–257 (1995)
29. Xu, Y., Géraud, T., Najman, L.: Context-based energy estimator: Application to object segmentation on the tree of shapes. In: ICIP, pp. 1577–1580. IEEE (2012)
30. Xu, Y., Géraud, T., Najman, L.: Morphological Filtering in Shape Spaces: Applications using Tree-Based Image Representations. In: Proc. of ICPR, pp. 485–488 (2012)
31. Zana, F., Klein, J.: Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. ITIP 10(7), 1010–1019 (2001)

Robust Keypoint Detection Using Dynamics

Miguel Angel Cataño[1] and Juan Climent[2]

[1] Pontificia Universidad Catolica del Peru, Peru
[2] Universitat Politecnica de Catalunya, Barcelona Tech, Spain
juan.climent@upc.edu

Abstract. In this paper we present a keypoint detector based on the bimodality of the histograms of oriented gradients (HOGs). We compute the bimodality of each HOG, and a bimodality image is constructed from the result of this bimodality test. The maxima with highest dynamics of the image obtained are selected as robust keypoints. The bimodality test of HOGs used is also based on dynamics. We compare the results obtained using this method with a set of well-known keypoint detectors.

Keywords: dynamics, test of bimodality, keypoint detection, histograms of oriented gradients.

1 Introduction

Keypoints of an image are widely used in many computer vision tasks, such as object recognition, robot navigation, tracking, or image retrieval. They are points of the image that must be easy to find, highly discriminative from other image points, and their location must be geometrically stable under different transformations of the image. Many keypoint detectors can be found in literature, i.e.: [9], [28], [14], [18], [1], ... having each one different advantages or drawbacks. Schmid [30] and Juan [13] reported comprehensive surveys about the performance of different keypoint detectors.

The Histograms of Oriented Gradients (HOGs) have mainly been used as keypoint descriptors attempting to capture the structural information of the keypoint obtained, taking into account the direction of the gradients in a portion of the image. The most popular one is the SIFT descriptor, proposed by Lowe [17]. Mikolajczyk and Schmid [21] presented the Gradient Location and Orientation Histogram descriptor (GLOH).

Instead of using HOGs as keypoint descriptors, in [4], they present a new method for keypoint detection based on the bimodality of HOGs. The bimodality level of a HOG gives very useful information about if a pixel can be considered as a robust keypoint. Concretely, they showed in [4] that highly bimodal HOGs identify keypoints with excellent discriminability. They build a bimodality image assigning a bimodality score to the center pixel of fixed size cells. Peaks of the bimodality image are candidates for being keypoints.

The keypoints are selected locating the relative maxima of the bimodality image. Mathematical morphology tools can help in this stage. The bimodality

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 402–412, 2013.

image makes possible to establish a hierarchy of keypoints based on their saliency. The contrast extinction values [33] of the obtained local maxima can be used to select the most significative keypoints. The dynamics of the maxima [8] of the bimodality image has also shown to be an excellent criterion for evaluating keypoint significance.

Another question that arises is which test of bimodality should be used. There are six popular test that are often used to measure the bimodality of a data set, namely, Variance Reduction Score (VRS) [11], Weighted Variance Reduction Score (WVRS) [11], DIP test [10], Kurtosis [7], Likelihood Ratio Test (LRT) [36], and Bimodality Index (BI) [35]. All these tests of bimodality have some drawbacks measuring the bimodality of HOGs obtained from a real image, mainly due to the fact that HOGs are cyclic, and the extension of these tests to cyclic histograms is not trivial. Furthermore, they present a remarkable lack of linearity to subtle histogram distortions. To overcome these problems, mathematical morphology offers again some useful tools. In [5] is presented a new test of bimodality based on the dynamics of the histogram extrema. In section 3 we give some details about this test.

We present a keypoint detector, based on the bimodality of HOGs. A bimodality image is so obtained. The keypoints are determined by the regional maxima having highest dynamics values. The bimodality test of HOGs is also based on dynamics. In the results section we compare the results obtained using our method with the most popular keypoint detectors.

2 Preliminaries Keypoint Detectors

Hessian Detector. One of the first keypoint detectors was proposed by Beaudet [2] based on Hessian matrix.

$$H = \begin{bmatrix} I_{xx}(x,\sigma) & I_{xy}(x,\sigma) \\ I_{xy}(x,\sigma) & I_{yy}(x,\sigma) \end{bmatrix} \tag{1}$$

Where $I_{xx}(x,\sigma)$, $I_{xy}(x,\sigma)$, and $I_{yy}(x,\sigma)$ are the second-order Gaussian smoothed image derivatives. It contains shape information by describing how the normal to an isosurface changes. Particularly interesting are the filters based on the determinant and the trace of the matrix H [32]. It has also been used to detect blob-like structures in the image.

Harris Detector. Hans Moravec [23], one of the first authors on define the word "Keypoint", proposed the basis for the detection of corners in an image. Chris Harris and Mike Stephens [9] proposed some improvements to the Moravec detector, making it the most used method nowadays, known as the Harris Detector.

Harris and Stephens noted that the response of the Moravec operator was anisotropic. They improved the localization performance by replacing the rectangular patches with Gaussian windows with a scale similar to the derivatives used. This method has proven to be one of the most reliable keypoint detectors.

The matrix used for Harris operator describes the gradient distribution in a local neighborhood of a point ($\mathbf{x}$):

$$M(\mathbf{x}) = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} I_x^2(\mathbf{x}, \sigma_D) & I_x(\mathbf{x}, \sigma_D) I_y(\mathbf{x}, \sigma_D) \\ I_x(\mathbf{x}, \sigma_D) I_y(\mathbf{x}, \sigma_D) & I_y^2(\mathbf{x}, \sigma_D) \end{bmatrix} \tag{2}$$

The eigenvalues of this matrix represent the principal signal changes in two orthogonal directions in a neighborhood around the central point. Based on this property, corners can be found as locations in the image where the image signal varies significantly in both directions, or for which both eigenvalues are large. In practice, Harris proposed to use the following measure for *cornerness*, which combines the two eigenvalues in a single measure and is cumputationally less expensive:

$$cornerness = det(M(\mathbf{x})) - k(trace(M(\mathbf{x})))^2 \tag{3}$$

Harris-Laplace. Mikolajczyk and Schmid [20] developed a scale invariant corner detector, referred to as Harris-Laplace. This method uses an automatic scale selection [22].

Automatic scale selection and the properties of the selected scales have been extensively studied by Lindeberg [16]. The idea is to select the characteristic scale of a local structure, for which a given function attains an extremum over scales. The ratio of the scales at which the extrema are found for corresponding points is the actual scale factor between the point neighborhood.

The Laplacian-of-Gaussian, LoG, finds the highest percentage of correct characteristic scales to be found.

$$|\mathrm{LoG}(\mathbf{x}, \sigma_n)| = \sigma_n^2 |L_{xx}(\mathbf{x}, \sigma_n) + L_{yy}(\mathbf{x}, \sigma_n)| \tag{4}$$

When the size of the LoG kernel matches the size of a blob-like structure, the response attains to an extremum.

The Harris-Laplace detector uses the scale-adapted Harris function (Eq. 3) to localize points in scale-space. It then selects the points for which the Laplacian-of-Gaussian, Eq. 4, attains a maximum over scale.

FAST Operator. Rosten and Drummong presented the operator FAST (Features from Accelerated Segment Test) [27].

It operates by considering a circular neighborhood around the keypoint candidate p. The detector classifies p as a corner if there exists a set of n contiguous pixels in the neighborhood which are all brighter than the candidate pixel I_p plus a threshold t, or all darker than $I_p - t$.

The second step to find corners employs the induction tree algorithm [26] and begins by selecting the x which yields the most information about whether the candidate pixel is a corner, measured by the entropy of K_p.

The entropy of K_p is:

$$H(P) = (c + \overline{c}) \log_2(c + \overline{c}) - c \log_2 c - \overline{c} \log_2 \overline{c} \tag{5}$$

$$\begin{aligned} \text{where } c &= |\{p|K_p \text{ is true}\}| \quad \text{(number of corners)} \\ \text{and } \overline{c} &= |\{p|K_p \text{ is false }\}| \text{ (number of non corners)} \end{aligned}$$

DoG Detector. Lindeberg proposed that the derivatives of Gaussian can determine basic invariance properties for first stages of image analysis [15]. Therefore, the scale space of an image is defined as a function, $L(x, y, \sigma)$, that is obtained from the convolution of a variable-scale Gaussian, $g(x, y, \sigma)$.

In order to detect stable keypoint locations in scale space efficiently, Lowe proposed in [17] the use of scale-space extrema in the difference-of-Gaussian (DoG) function convolved with the image, $D(x, y, \sigma)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k:

$$D(x, y, \sigma) = (g(x, y, k\sigma) - g(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \qquad (6)$$

The features obtained are invariant to image scale and rotation. We highlight that Lowe uses HOGs to assign a consistent descriptor to each keypoint.

Fast-Hessian Detector. Herbert Bay proposed the operator SURF [1] (Speeded Up Robust Features). It is based on the Hessian matrix because of its good performance in computation time and accuracy. However, rather than using a different measure for selecting the location and the scale (as in the Hessian-Laplace detector [20]), it relies on the determinant of the Hessian for both.

For optimization, Bay uses the approximate second order Gaussian derivatives, because it can be evaluated very fast using integral images.

Bay builds a scale spaces growing the size of the filter. So, use filters of size 9x9, 15x15 21x21, 27x27, etc. In order to localize interest points in the image and over scales, a nonmaximum suppression in a 3D neighborhood is applied.

Hessian-Laplace detector The Hessian-Laplace detector, proposed by Mikolajczyk and Schmid [22], is similar in spirit as the Harris-Laplace, except that it starts from the determinant of the Hessian rather than the Harris corners. This turns the methods into viewpoint invariant blob-detector.

3 Test of Bimodality Using Dynamics

The dynamics, introduced by Grimaud [8], is a mathematical morphological measure associated to each regional extrema of an image. This measure is a powerful tool to quantify the saliency of a maximum or minimum. It has usually been used as a concise and powerful measure of contrast for the identification of regions of interest in the image, but in our work we use dynamics to quantify the bimodality of an histogram.

The dynamics of a regional minimum is defined as the minimum height we have to climb to reach another one with strictly higher dynamics, the climb being the difference in altitude between the highest point of the path and the regional minimum under study. Formally: Let M be a regional minimum of a function f. The dynamics of M is defined as:

$$\min\left\{ \max_{s\in[0,1]} \{f(\gamma(s)) - f(\gamma(0))\} \;\middle|\; \gamma : [0,1] \to \Re^2, f(\gamma(1)) < f(\gamma(0)), \gamma(0) \in M \right\} \tag{7}$$

where γ is a path linking two points.

Note that the dynamics are not defined for the global minimum of the image, as there is no pixel with strictly lower altitude. However, it is assumed that the image f has a global minimum on its boundary. This makes possible the valuation of the global minimum inside the domain of definition of f.

Grimaud [8],presents the dynamics of regional minima instead of maxima. However, we need the dynamics of the maxima of the histogram to compute the bimodality. Dynamics of a regional maximum M is the minimum height that we should fall, to reach another regional maximum higher than M. Figure 1 illustrates this idea. To determine the dynamics of M_2 (DM_2), there are two higher maxima, M_1 and M_5, but it is necessary to climb down h_1 in the first path and h_2, in the second. Being $h_2 < h_1$, the second path (dotted) is preferred, and $DM_2 = h_2$.

The use of regional maxima agrees with the characteristics of the Max-tree. A solution according to the original formulation, could also be modeled with the Min-tree of the negative image.

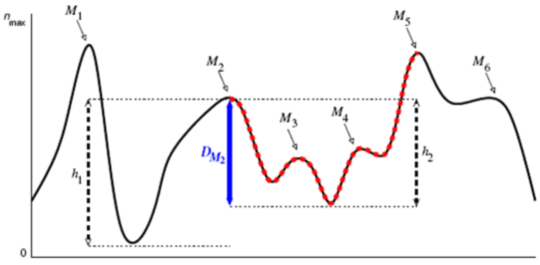


Fig. 1. The dynamics of the maxima

Once we have the values of the two highest dynamics, dyn_1 and dyn_2, the bimodality of the histogram is computed using equation 8

$$BM = \frac{dyn_1 \cdot dyn_2}{\sqrt{dyn_1^2 + dyn_2^2}} \tag{8}$$

The product of the dynamics is divided by the Euclidean norm for normalization purposes. The algorithm for extracting the dynamics is based on the watershed algorithm proposed by Vincent and Soille [34]. It consists in flooding from the minima, level by level, until water from a catchment basin meets water from another one. The meeting point between two basins is a saddle point, and this is the point where we can compute the dynamics of one of the two basins: the basin with the lowest value floods the other one, and the dynamics of the basin with the lowest minimum is equal to the grey-value of the saddle point minus the grey-value of the minima.

Organizing the image as a set of regions enables a faster implementation of dynamics. In this sense, the Component Tree[12,19] is a structure formed by the decomposition of a grey-level image by thresholds, and very useful for dynamics determination[3] . It is especially interesting because it requires only the adjacency definition for its construction, and there exist several quasi-linear time algorithms for constructing it[24].

Max-tree refers to the algorithm introduced by Salembier[29] for an efficient implementation of the component tree. Silva and Lotufo describe in [31] an efficient computation of the dynamics, based on the incremental determination of attributes from the max-tree construction, in linear time.

4 Results

In this section, first we describe the process details for detecting the keypoints, then we present the repeatability as the standard measure to test the robustness of a keypoint detector. Finally we compare the repeatability results of our method with the ones obtained with the most popular keypoint detectors.

A HOG is computed for the corresponding cell of every image pixel and after that, we compute our bimodality test based on the dynamics for every HOG. Then, every central pixel of each cell receives its corresponding bimodality score, and a bimodality image is built. Peaks of the bimodality image are candidates for being keypoints. We select the keypoints locating the relative maxima of the bimodality image. Figure 2b shows the bimodality image of the synthetic image 2a.

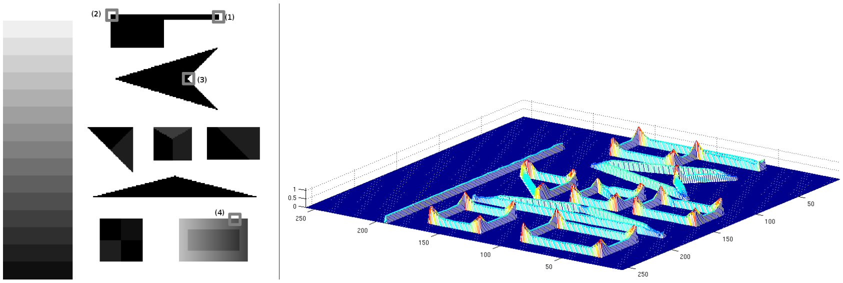


Fig. 2. (a) A synthetic image (b) A 3D representation of the bimodality image

The HOGs defined in [6] are blocks built from histograms of orientations obtained from overlapping cells, and they were originally used to detect people in scenes. We work just with the histograms of gradient orientation of single cells, thus, there is no need to work with blocks since the objective is not to detect objects but simple keypoints. Our histograms are also constructed differently from the SIFT descriptors, which are based on HOGs too. We compute the histograms at a single scale and without dominant orientation alignment. Therefore, computing the histograms in our application becomes an easy task.

We first compute the gradient module and orientation images. For every pixel in the orientation image, we construct a histogram of orientations over local spatial windows called cells. Each pixel in the cell votes for its corresponding orientation value. The weighted votes are accumulated into the orientation bins.

Histograms computation is speeded up using the integral histograms presented in [25]. The integral histogram is computed in linear time. Once we have the integral histogram, for each cell a HOG is computed with just four additions.

Once the HOGs have been computed, the next step is to obtain the bimodality image. We have modified the original algorithm for extracting the dynamics to make it able to work with cyclic histograms. As mentioned earlier, the dynamics of the absolute maximum of an image is computed considering that the image boundary has a virtual higher value. Since we are dealing with cyclic histograms, we cannot make this assumption because a circular structure has no boundaries. Thus, the global maximum of the histogram will be always considered the maximum with highest dynamics, and quantified with its height.

The other variation introduced to the original algorithm also concerns the circularity of HOGs. We have modified the original algorithm in order to avoid the zero crossing discontinuity. We have considered the circularity of the histogram in the flooding process, and taken into account that a saddle point can split two domes whose maxima are at both sides of the histogram origin. The zero crossing could also split a single dome in two, giving as a result two regional maxima, one at each side of the histogram origin. Due to the nature of the flooding process, this modification of the algorithm does not translate into any additional computational cost.

In order to test the robustness of the keypoints detected, we change the image point of view with a projective transform (see figure 3).

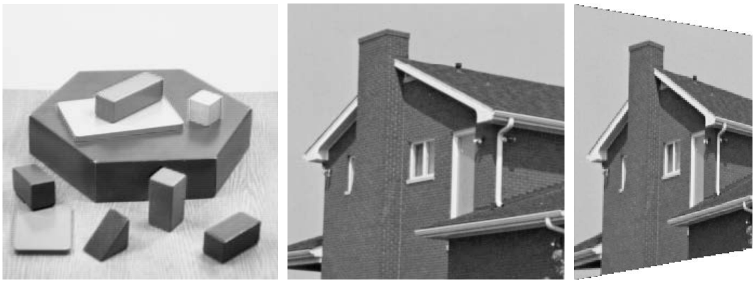


Fig. 3. (a) Image blocks, (b) Image house, (c)Image house projected 50°

Keypoints detected in one image should also be detected at approximately corresponding positions in the subsequent image transformations. The most commonly used criterion to prove that the keypoint detection is stable to image transformation is repeatability [30]. The repeatability rate, *r(I,ψ(I))*, is defined as the number of keypoints present in both the original I and the transformed image *ψ(I)*, with respect to the total number of detected keypoints. Keypoints that are not present within a certain neighborhood in both images, decrease the repeatability measure. We have computed the repeatability *$r(\psi_\alpha(Keyp(I)),Keyp(\psi_\alpha(I)))$* for all projective transforms ψ_α from α=0°to α=70°. *Keyp(I)* is an image containing the keypoints of image *I*. Fig. 4 shows a block diagram of the process. This process is repeated for all different keypoint detectors.

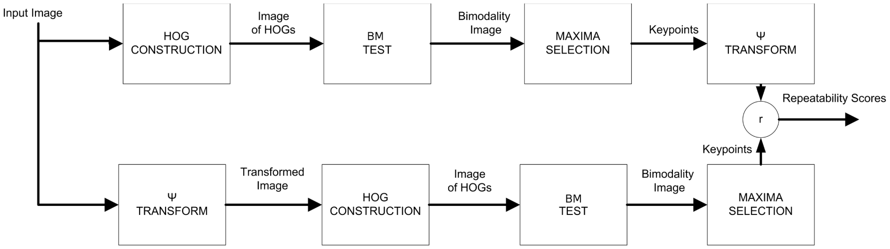


Fig. 4. Process for measuring repeatability

The repeatability rates of our method are compared with the ones obtained from the most popular keypoint detectors. Concretely, we have compared our keypoint detector based on HOGs with Harris, Harris-Laplacian, Hessian-Laplacian , FAST, SURF, and SIFT operators. For a fair evaluation, these should not be a significant discrepancy among the amount of keypoints detected by these algorithms. For this propose, we have fixed the parameters of all algorithms to give a constant amount of 100 keypoints. Therefore, there is no need to raise the issue about considering absolute o relative repeatability. For our method we have selected the 100 maxima having highest dynamics. Furthermore, the parameters of all other keypoint detectors have been tuned to achieve the highest repeatability for every tested set of transformed images. Our algorithm works with unsigned gradient, 180 bin histograms, and 17x17 pixels cells. We show the results for the synthetic image (fig. 2a), the image blocks (fig. 3a) and the image house (fig. 3b). All images have progressively been distorted using projective transforms (fig. 3c).

Figure 5 shows the repeatability of all methods for projective transformations of the image under different viewpoints from 0° to 70°. The repeatability obtained using HOGs is higher than most of the other methods for most of the different transforms.

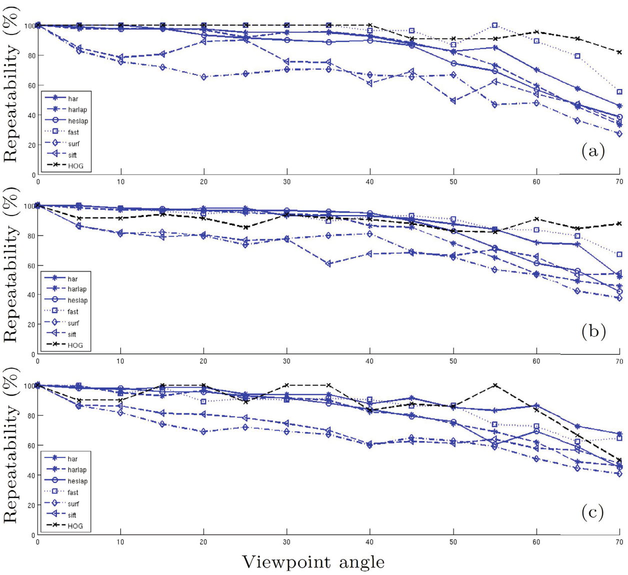


Fig. 5. Repeatability for the images:(a)synthetic, (b)blocks and (c)house. Images are transformed with a projection from 0° to 70°.

5 Conclusion

HOGs with high or low entropy, usually unimodal histograms, never correspond to discriminative keypoints. This is due to the fact they are too similar to their neighbor HOGs. The higher the unimodality of the HOG, the less discriminative will be the keypoint. In this paper we have presented the use of histograms of oriented gradients in the keypoint detection stage. The limitations of existing solutions make the keypoint detection too dependent on image variations, and their localization inaccurate. The method presented has shown to be more robust to projective transformations of the image. Mathematical morphology offers some interesting tools to achieve our goals. First, the test of bimodality based on dynamics overcomes the main drawbacks of existing tests. Furthermore, the dynamics of the maxima of the bimodality image has shown to be an excellent measure for evaluating the keypoint significance. So far, we have considered only fixed-size cells to compute the histograms, however, cell size has a determinant effect on the image scale where keypoints are detected. In this work we have not considered working at different scales. The research for the extension of this method to the scale space is currently in process.

Acknowledgments. This research was partially supported by Consolider Ingenio 2010, project (CSD2007-00018) and CICYT project DPI2010-17112.

References

1. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding 110(3), 346–359 (2008)
2. Beaudet, P.R.: Rotationally invariant image operators. In: Proceedings of the International Joint Conference on Pattern Recognition, vol. 579, pp. 579–583 (1978)
3. Bertrand, G.: On the dynamics. Image Vision Comput. 25(4), 447–454 (2007)
4. Cataño, M.A., Climent, J.: Keypoint detection based on the unimodality test of hogs. In: Bebis, G., et al. (eds.) ISVC 2012, Part I. LNCS, vol. 7431, pp. 189–198. Springer, Heidelberg (2012)
5. Cataño, M.A., Climent, J.: A new morphological measure of histogram bimodality. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) CIARP 2012. LNCS, vol. 7441, pp. 390–397. Springer, Heidelberg (2012)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893 (June 2005)
7. DeCarlo, L.T.: On the meaning and use of kurtosis. Psychological Methods 2(3), 292–307 (1997)
8. Grimaud, M.: New measure of contrast: the dynamics, vol. 1769, pp. 292–305. SPIE (1992)
9. Harris, C., Stephens, M.: A combined corner and edge detector, Manchester, UK, vol. 15, pp. 147–151 (1988)
10. Hartigan, J.A., Hartigan, P.M.: The dip test of unimodality. The Annals of Statistics 13(1), 70–84
11. Hellwig, B., Hengstler, J., Schmidt, M., Gehrmann, M., Schormann, W., Rahnenfuhrer, J.: Comparison of scores for bimodality of gene expression distributions and genome-wide evaluation of the prognostic relevance of high-scoring genes. BMC Bioinformatics (1), 276+
12. Jones, R.: Connected filtering and segmentation using component trees. Comput. Vis. Image Und. 75(3), 215–228 (1999)
13. Juan, L., Gwun, O.: A comparison of sift, pca-sift and surf. Image Processing 3(4), 143–152 (2009)
14. Kadir, T., Brady, M.: Saliency, Scale and Image Description. International Journal of Computer Vision 45(2), 83–105 (2001)
15. Lindeberg, T.: Scale-space theory: A basic tool for analysing structures at different scales. J. of Applied Statistics 21(2), 224–270 (1994)
16. Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision 30, 79–116
17. Lowe, D.G.: Object Recognition from Local Scale-Invariant Features. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (August 1999)
18. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
19. Mattes, J., Richard, M., Demongeot, J.: Tree representation for image matching and object recognition. In: Bertrand, G., Couprie, M., Perroton, L. (eds.) DGCI 1999. LNCS, vol. 1568, pp. 298–312. Springer, Heidelberg (1999)
20. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: Proceedings. Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 1, pp. 525–531 (2001)

21. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
22. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. International Journal of Computer Vision 60, 63–86
23. Moravec, H.P.: Obstacle avoidance and navigation in the real world by a seeing robot rover. Ph.D. thesis, Stanford, CA, USA, aAI8024717 (1980)
24. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE T. Image Process. 15(11), 3531–3539 (2006)
25. Porikli, F.: Integral histogram: A fast way to extract histograms in cartesian spaces. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 829–836 (2005)
26. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1(1), 81–106 (1986)
27. Rosten, E., Porter, R., Drummond, T.: Faster and better: A machine learning approach to corner detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(1), 105–119 (2010)
28. Rosten, E., Drummond, T.W.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
29. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. IEEE T. Image Process. 7(4), 555–570 (1998)
30. Schmid, C., Mohr, R., Bauckhage, C.: Comparing and evaluating interest points. In: Sixth International Conference on Computer Vision, pp. 230–235 (January 1998)
31. Silva, A.G., de Alencar Lotufo, R.: Efficient computation of new extinction values from extended component tree. Pattern Recogn. Lett. 32(1), 79–90 (2011)
32. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: A survey. Foundations and Trends in Computer Graphics and Vision 3(3), 177–280 (2007)
33. Vachier, C., Meyer, F.: Extinction value: a new measurement of persistence. In: IEEE Workshop on Nonlinear Signal and Image Processing, vol. 1, pp. 254–257 (1995)
34. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE T. Pattern Anal. 13(6), 583–598 (1991)
35. Wang, J., Wen, S., Symmans, W.F., Pusztai, L., Coombes, K.R.: The bimodality index: a criterion for discovering and ranking bimodal signatures from cancer gene expression profiling data. Cancer Informatics 7, 199–216 (2009)
36. Wolfe, J.H.: Pattern Clustering by Multivariate Mixture Analysis. Multivariate Behavioral Research 5, 329–350 (1970)

A Granulometry Based Descriptor for Object Categorization

Arnaldo Câmara Lara and Roberto Hirata Jr.

University of Sao Paulo, Institute of Mathematics and Statistics
Rua do Matao, 1010, 05508-090, Sao Paulo, Brazil
alara@vision.ime.usp.br, hirata@ime.usp.br

Abstract. The progress in the area of object recognition in the last decade is impressive. The literature reports new descriptors, new strategies, new ways to combine descriptors and classifiers and new problems in a so fast pace that it is hard to follow the whole area. A recent problem in the area is the fine-grained categorization. In this work, to address this problem, we propose a descriptor based on the application of morphological granulometries in the map of edges of an image. This descriptor is used to characterize the distribution of lengths and orientations of edges and to build a model for generic objects. We also propose a new spatial quantization with an arbitrary number of levels and divisions in each level. This quantization is so flexible that adjacent regions may have overlapping areas to avoid breakages in the structures that are near the border of the regions as it happens in the traditional spatial pyramids. Both approaches are used in a challenging and recent object recognition problem, the categorization of very similar classes. The proposed descriptor was used along with other descriptors and the overall performance of our solution to this problem was about 8% better than other work using the bag-of-words approach reported in the literature. Our descriptor showed a result 12% better when compared to the results of other edge-related descriptor in the categorization of very similar classes.

Keywords: Granulometry, Application, Spatial Quantization, Descriptor, Edges, Object Categorization.

1 Introduction

Object recognition is a Computer Vision (CV) area that studies the problem of recognizing objects in an image or video frame. A typical problem in this area is to label an object in an image as a member of a set of known classes. That problem is known as: *object categorization.* It is a simplification of the original problem because it constraints the labeling to a finite set of known classes. Object categorization datasets have labeled images and they are used to train a classifier or a set of classifiers to produce a model for each different class. A sample is classified by comparing it to a set of models and assigning to it the most likely label. When the classes of a dataset are too similar, the problem is known as *fine-grained categorization* [10].

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 413–424, 2013.

In this work, we propose a descriptor named: "Granulometry Based Descriptor of Edges", or GRABED for short. The descriptor is based on morphological granulometries applied to the map of edges of the images that are used to build the models for the objects in the dataset. Granulometries are usually applied to binary or grayscale images and they measure the distribution of sizes and shapes (binary case), or the distribution of textures and shapes (grayscale case). We propose the application of families of morphological granulometries in a different way, i.e., to the map of edges of an image to characterize the objects in it. This descriptor has been tested in an object categorization problem using a public database with results similar to the state-of-the-art descriptors traditionally used in this kind of problem. We also propose a spatial quantization with some flexibilities when compared to other quantizations, for instance, spatial pyramids [1]. The new spatial quantization may have: an arbitrary number of levels, regions (in each level) and weights (for each level). A direct consequence is that a region of a level may not be a partition of the regions of the prior level. The proposed quantization has also a parameter that controls an overlapping area in the border of the regions of a given level. This flexibility helps to avoid breakages of structures (edges) that are near the border of the regions and improves the performance of models created using this spatial quantization strategy.

Both proposals were used in a novel solution for a fine-grained object categorization problem using a challenging public database. We built a class-specific model using a combination of descriptors including the GRABED descriptor. The overall performance of our technique technique was 8% better than other work that also uses the Bag-of-Words (BoW) approach in the same problem and dataset.

The text is organized in the following way: this section introduces the work and the next section describes the object recognition problem. Section 3 briefly reviews morphological granulometry. Section 4 describes the novel descriptor (GRABED). The experiments and results are reported in Section 5. Conclusions and final remarks are given in Section 6.

2 Object Recognition

Bag-of-Words (BoW) is one of the most popular approaches used to deal with the object recognition problem [19]. It achieves good results in the categorization process at a reasonable computational cost. BoW uses local descriptors that are clustered, usually using k-means [4]. The central point of each cluster is a codeword or visual word. The collection of all codewords forms a visual dictionary and the final descriptor of an object is a histogram of codeword frequencies. No spatial information is kept in the final descriptor. Spatial pyramids, proposed in [1], is a very popular technique that adds some type of spatial information to the orderless BoW approach improving its results [21]. A spatial pyramid consists of partitioning an image in increasingly smaller sub-regions and computing BoW in each sub-region. It can be viewed as an extension of the BoW approach because a spatial pyramid may have just one level.

Fine-grained object categorization is a problem of object categorization when the classes are very similar, for instance: species of different birds [11]. This problem is very difficult. So far, there is no effective approach to deal with the problem yet [10]. The traditional BoW, even in conjunction with spatial quantization does not present good results. BoW is a generalization technique and during its application some subtle and discriminant characteristics are lost. Besides that, BoW descriptors are not properly localized even with spatial quantization that is a rough localization of descriptors (that is also a kind of generalization). Some methods use human annotations in the training step to help building more accurate models for the classes [12]. A disadvantage of this technique is the high cost of human annotations. In [10], a template matching approach is used to solve a birds' categorization problem obtaining good results. A large number of image templates are generated in random positions and dimensions for each image in the training set. A response map is computed comparing the test image in different scales with each random generated template. Seven values are extracted from each response map and all these values are concatenated in a template matching based representation of the test image. The size of the feature vector is 882000.

3 Granulometry

Granulometry has been firstly proposed by Matheron [6] to characterize images by the distribution of a shape and its sizes. The idea is to think of an image as a collection of grains that are sieved by a collection of screens of increasing sizes. The first sieve holds the grains that are larger than the smaller screen size. The second sieve holds the grains that are larger than the second smallest screen size (that is larger than the first screen size) and so on until no more grains are held. Granulometries are usually implemented by successive applications of morphological openings (or closings) [8] of increasing sizes. The quantification of the rate that grains pass through the mesh of sieves produces a numeric distribution that characterizes the texture and shape of the image [7].

In this work we will attain ourselves to granulometries of morphological openings γ_{tB}, where B is a convex structuring element (SE) and it is known as generator of the granulometry [8], $t \geq 0$ and $tB = \{t\mathbf{b} | \mathbf{b} \in B\}$.

Let $\Omega(t)$ define the area, in the binary case, or the volume, in the grayscale case, that remains in the image after the application of the opening by tB. $\Omega(0)$ is the area, or volume of the map of edges. $\Omega(t)$ is usually called *size distribution*. The *normalized size distribution* scales the values of $\Omega(t)$ to $[0, 1]$ and it measures how much of the total area or volume are vanished from the original image until t: $\Phi(t) = 1 - \frac{\Omega(t)}{\Omega(0)}$. The derivative of $\Phi(t)$ is a probability density function and it is known as the *pattern spectrum* of the image. The pattern spectrum can be calculated globally or locally using a window [8]. In the discrete case, pattern spectrum $PS(t)$ can be calculated as:

$$PS(t) = \Phi(t) - \Phi(t-1). \quad (1)$$

4 GRABED Descriptor

GRABED descriptor is based on morphological granulometries applied to the map of edges of an image to compute class-models and later use them in object categorization problems. The descriptor measures the distribution of orientations and lengths of the edges of an image using openings by linear SEs. This last characteristic differs it from Histogram of Oriented Gradients (HOG) descriptor [9], another gradient-based descriptor, because HOG only calculates local histograms of orientations but not lengths.

Basic Descriptor. The descriptor is calculated according to the following steps:

1. A parameter R defines how many different angles are going to be used to rotate the linear SEs between 0 and 180 degrees producing the set of orientations $O = o_1, o_2, o_3, ..., o_R$. As showed in [8], it not trivial to define all possible directions of a 3 pixels-length linear segment in a discrete space. In segments longer than that, it is practically impossible. So, we decided to choose a fixed number of orientations to use. The higher the number of orientations, the best is the accuracy but also, the higher is the computational cost. It is a trade-off in the choice of parameters.
2. A family of linear SEs is produced for each orientation defined in the previous step. The length of the first SE, in pixels, is equal to L_0 (a parameter of the descriptor). The next SE is twice longer. The third SE is twice longer than the second SE and so on. The last SE of the family has length longer than half of max_axis. The length of the i-th SE of a family of SEs, $len(i)$, can be written as:

$$len(i) = \begin{cases} L_0 & \text{if } i = 1; \\ 2 * len(i-1) & \text{if } i > 1 \text{ and } len(i-1) < \frac{max\_axis}{2}. \end{cases} \qquad (2)$$

   To strictly satisfy the granulometric properties, the family of line segments must be extended k pixels at a time [8]. However, to reduce the number of openings and the computational cost of the descriptor, we double the length of the linear SEs at a time.
3. The map of edges of a given image is produced using Canny's method [14]. In a prior work [13], empirical experiments showed that Canny's is more stable than other methods to produce the map of edges.
4. For each family of linear SEs produced in Step 2, a set of openings are applied to the map of edges using each SE:

$$\gamma_{B_{i,j}}(X), \qquad (3)$$

   where $B_{i,j}$ is the i-th SE of the family of SEs with orientation $o_j \in O$. So, $\Omega(i)$ is calculated counting the number of pixels in the result of the i-th opening. $\Omega(0)$ is the number of pixels in the original map of edges of the image.

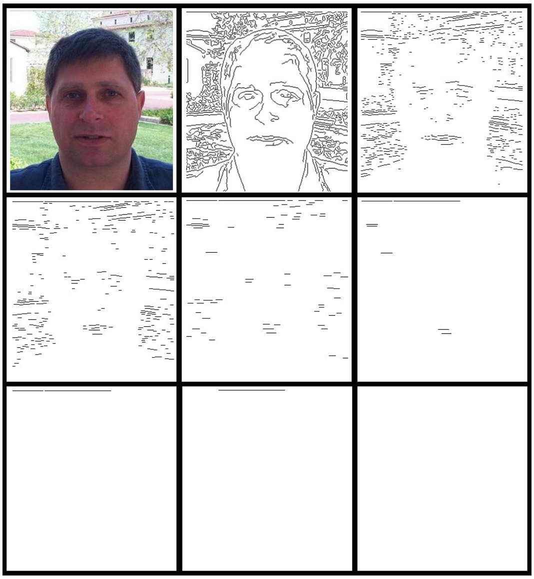


Fig. 1. Steps to compute the pattern spectrum for an image of class Faces_easy. The family of SEs used has 0° of orientation. From top to bottom, left to right, the images are: original image, map of edges, results of morphological opening using a linear SE with 2, 4, 8, 16, 32, 64 and 128 pixels of length, respectively.

5. For each family of linear SEs and using $\Omega(i)$ calculated in the previous step, PS(i) is calculated using Equation 1:

$$PS(i) = \Phi(i) - \Phi(i-1) = 1 - \frac{\Omega(i)}{\Omega(0)} - (1 - \frac{\Omega(i-1)}{\Omega(0)}) = \frac{\Omega(i-1) - \Omega(i)}{\Omega(0)}. \quad (4)$$

6. The final descriptor is a concatenation of all PSs calculated in the previous steps.

Figure 1 shows the computation of the PS using a family of SEs for an image of class Faces_easy of Caltech-101 database. The family of SEs is composed by linear SEs with 0° of orientation and initial length of 2 pixels. From top to bottom, left to right, the figure shows the original image, the map of edges calculated using Canny's and the results of morphological opening using a linear SE of 2, 4, 8, 16, 32, 64 and 128 pixels.

Multiscale. The GRABED descriptor is calculated in a multiscale approach. A parameter S indicates the number of scales to be used. The first scale uses the image in the original size and a parameter F controls the resize factor. The idea is to model morphological structures of different scales from details to the general shape of the object. For each scale and orientation, the set of PSs described

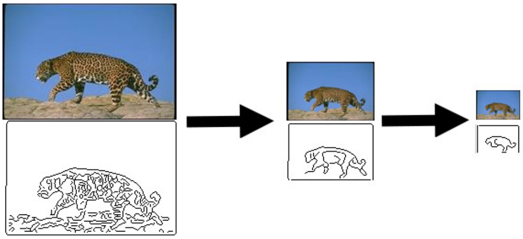


Fig. 2. An example of an image of class Leopards in 3 different scales

in the basic descriptor is computed. The resulting descriptor of the multiscale approach is the concatenation of descriptors calculated for each scale. Figure 2 shows an image of class Leopards. It was used $S = 3$ and $F = 0.5$.

Spatial Quantization. Spatial pyramids (SPM) add spatial information to the BoW approach improving its overall results [1]. We propose a new spatial quantization strategy that has less restrictions in relation to the original proposal. Flexible Spatial Quantization (FSQ), proposed here, has an arbitrary number of levels and regions in each level. Besides that, for each level, the regions do not need to be partitions of the previous level as in SPM and they can overlap. Given a FSQ with L levels, a weight $\omega_i, i \in [1..L]$ is applied to the descriptors of level i. The constraints for the weights are $\omega_i \geq 0$ and $\sum_{i=1}^{L} \omega_i = 1$. Because they can overlap, a parameter V controls how many pixels of the border of a region advances over its neighboring region. The motivation for this redundancy in the representation is to reduce the breakages of important structures that happens to be in the frontiers of the regions and that may be important for the discrimination of the image. The final descriptor is the concatenation of the descriptors for each region of each level. Figure 3 shows (upper left corner) an original image of the class umbrella of Caltech-101 database and its map of edges in the upper right corner. In the bottom left, it shows a spatial quantization of the map of edges with one level and 3 regions in this level. No overlapping area was set. In bottom right, it shows a spatial quantization with the same divisions but with an overlapping area. Each border moves over the adjacent region in 30 pixels. As it can be seen, this helps to preserve the structure of edges present in border region.

4.1 Class-Specific Model for Fine-Grained Categorization

The proposed descriptor and spatial quantization are used in a new method for fine-grained object categorization. We used a subset of 13 classes of a birds database [11]. Our fine-grained object categorization has the following steps:

1. ***Initialization***: using a similar approach as the one used in [15], new images are generated from the original ones (in the training set) through geometric transformations. Images are resized by a factor F and in S different scales.

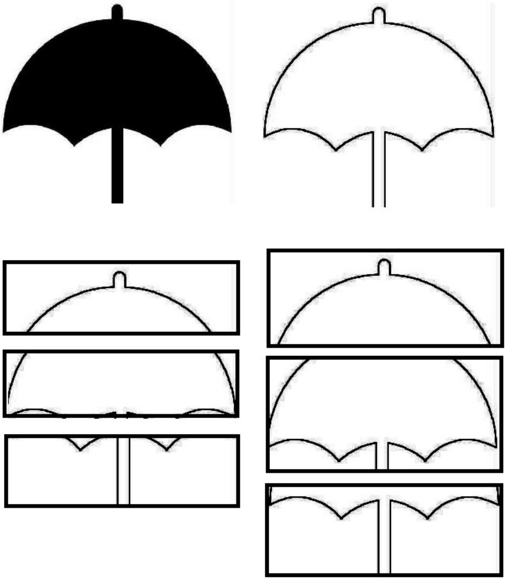


Fig. 3. From top to bottom, left to right, it shows the original image, the map of edges calculated using Canny's, spatial quantization without overlapping area and spatial quantization with an overlapping area

A variable M specifies if the image is mirrored, or not, and a parameter R specifies the number of different angles of rotation applied. The total of generated images are $S \times 2 \times R \times len(training\_set)$, where $len(training\_set)$ is the length of the training set. The set of descriptors to be used in the building of models can be any combination of the following descriptors: PHOW [16], SURF [3], LBP [22], RGB Histogram [8], GRABED and PHOG [5].

2. ***Generation of descriptors***: A visual dictionary for the BoW approach can be built if PHOW, or SURF, is in the set of descriptors. For each visual dictionary, a parameter N is used to cluster the descriptors into N visual words using k-means. Then, a matrix of descriptors for the images in the training set is built [19]. The spatial quantization strategy described previously is used in the generation of all descriptors.
3. ***Building of models***: two different types of models are built for each used descriptor:
   (a) *Model level 1*: the 13 classes of the birds dataset can be grouped in two different groups. Seven classes of genus Vireo and six classes of woodpeckers. Feature vectors of birds of genus Vireo are labeled as 1 and feature vectors of woodpeckers are labeled as 2. A model is built using the labels of these two groups of classes.
   (b) *Model level 2*: 13 models are built, one model for each class, in a One-Vs-All, i.e., for each model, feature vectors of a class are labeled as 1 and all other feature vectors are labeled as 2.

Table 1. MAP metric for 5 used classes in 3 different configurations of GRABED descriptor

Configuration	MAP(%)
GRABED	82
GRABED + multiscale	89
GRABED + multiscale + FSQ	97

A total of 14 models for each descriptor are built in this step. A SVM using chi-squared kernel is used as the classifier. A cross-validation step using k-fold is used during the building of models to measure the confidence of each model [17]. We used a $k = 5$ for the k-fold cross-validation method in experiments. The precision for each model is calculated. Models that do not have a confidence greater than 0.75 for a level 1 model and greater than 0.3 for a level 2 model are eliminated. These values have been chosen empirically.

4. ***Testing***: For each sample in this step, it is compared to all remaining models. The outputs of the SVM classifier is translated into confidence for each label [23]. The probabilities are uniformly divided by the number of classes that are represented by each label. The final probability of a sample belonging to a class is the average of probabilities of all used models. The class with the highest probability labels the sample.

5 Experiments and Results

This section describes the experiments and shows the results of the proposed techniques.

To test the methods for object categorization, we have chosen five classes of Caltech-101 database [18]: BACKGROUND_Google, Faces, Faces_easy, Leopards and Motorbikes. Those are the five classes with the largest number of images in the database. We used 100 images for each class in the training set and 50 images for each class in the test set. The used parameters are $R = 12$, $L_0 = 2$ and $max\_axis = 500$ pixels. Table 1 shows the mean of average precisions (MAP) metric [20] for the 5 used classes using GRABED descriptor in 3 different configurations: GRABED alone, GRABED + the multiscale approach and GRABED + the multiscale approach + the spatial quantization. For the multiscale approach, $S = 3$ and $F = 0.5$. For the spatial quantization, a FSQ with 2 levels was used: one region in the first level and nine regions in the second.

Table 2 shows the results for each of the 5 classes of the descriptor using the same parameters for all classes. It uses 2 different scales and a FSQ with 1 region in the first level and 3-by-3 regions in the second level. Accuracy and Precision [20] are abbreviated to Acc. and Prec., respectively. The table also shows the recall.

Table 3 shows the best results for each class using its best setup. As it can be seen, each class has its ideal configuration. Therefore, the overall performance has a small improvement compared to Table 2.

Table 2. Result for each tested class using the same parameters

Class	Measures		
	Acc.(%)	Prec.(%)	Recall(%)
BACKGROUND_Google	91	90	92
Faces	98	96	100
Faces_easy	99	98	100
Leopards	99	98	100
Motorbikes	97	100	94
Average	97	97	97

Table 3. Best parameters for each of the 5 classes of GRABED descriptor

Class	Parameters				Measures		
	FSQ	Scales	R	L_0	Acc.(%)	Prec.(%)	Recall(%)
BACKGROUND_Google	1x1;3x3	3	12	2	91	87	96
Faces	1x1;4x4	3	12	2	100	100	100
Faces_easy	1x1;2x2	2	12	2	100	100	100
Leopards	1x1;3x3	2	12	2	99	98	100
Motorbikes	1x1;2x2	3	12	2	98	100	96
Average					98	97	98

Table 4 shows a comparison between GRABED and the other tested descriptors. The best parametrization for GRABED is used and compared to the best setup of parameters of the other tested descriptors.

5.1 Experiments in a Fine-Grained Categorization Database

We used a combination of descriptors (including GRABED) and FSQ in a new technique for fine-grained object categorization (Section 4.1) of birds species. The Caltech-UCSD Birds-200 2011 database [11], that is a very important database focused in fine-grained categorization, has been used for this experiment. A subset of the database with 6 species of Woodpeckers and 7 species of genus *Vireo*

Table 4. The comparative results of GRABED and the other tested descriptors: PHOG, PHOW, SURF and SIFT. Average values of the 5 classes, the best parametrization for each descriptor was used.

Descriptors	FSQ	Multiscales	BoW	Measures		
				Acc.(%)	Prec.(%)	Recall(%)
PHOG	Yes	Yes	No	98	97	99
PHOW	Yes	Yes	Yes	98	97	98
GRABED	Yes	Yes	No	97	97	97
SURF	No	No	Yes	88	86	93
SIFT	No	No	Yes	86	84	90

Table 5. Average precision (AP) for each tested class of Caltech-UCSD Birds-200 2011 database

Class	AP(%)	Descriptors
151	33.4	PHOW
152	58.8	GRABED, RGB Histogram, PHOW, PHOG
153	36.3	GRABED, RGB Histogram, PHOW, PHOG
154	21.2	GRABED
155	24.9	PHOW
156	20.3	GRABED, PHOW, PHOG
157	24.7	GRABED, RGB Histogram, PHOW, PHOG, LBP, SURF
187	46.4	GRABED
188	40.4	GRABED, RGB Histogram, PHOW, PHOG, LBP, SURF
189	56.5	GRABED, RGB Histogram, PHOW, PHOG, LBP, SURF
190	70.0	GRABED, RGB Histogram, PHOW, PHOG, LBP, SURF
191	43.9	GRABED, RGB Histogram, PHOW, PHOG, LBP, SURF
192	45.8	GRABED, RGB Histogram, PHOW, PHOG, LBP, SURF
MAP	40.2	

has been used. This same subset was also used in other works [10]. Categorization of bird's species is a challenging task because bird's visual appearance can vary significantly because of: their different poses, different situations as flying, in water or in the nest. Besides that, there are also differences between males and females. No manual annotation was used during these experiments except the bounding-box during the training step. In the initialization step, it was used 3 different scales, 9 different rotations between $-20°$ and $+20°$. Thus, for a total of 389 images, 21006 images are generated for training. The visual dictionaries for PHOW and SURF descriptors were built using 600 codewords.

Table 5 shows the best average precision (AP) for each tested class. GRABED and PHOW are present in 11 of 13 best models. The obtained MAP for all classes was 40.2. It is about 8% better than another work using BoW approach reported in literature [12]. Using just one descriptor to classify all 13 classes, GRABED descriptor obtained a MAP of about 12% better than PHOG, another gradient related descriptor that also use a spatial quantization.

5.2 Discussion

GRABED showed results similar to PHOW and PHOG in the experiments reported in Section 5. PHOW is reported in literature as one of the best descriptors for object categorization [2]. It showed also the importance of multiscale approach and the spatial quantization. The basic GRABED descriptor obtained about 82% of MAP. Multiscale approach improved the overall results in about 8.5% and the combination of multiscale approach and spatial quantization improved the MAP of the classification in about 18%.

As GRABED characterizes different features when compared to PHOW and because they have similar performances in our tests, we tested the combination of both descriptors and some others descriptors to solve the bird's categorization problem. The obtained results, MAP of 40.2%, outperform other results using the same BoW approach. Our technique uses 2-levels classifications, class-specific combination of descriptors and different types of models for each descriptor. The proposed method is incremental when compared to the original BoW approach and it is not linked to a specific database. It can be easily adapted to solve other fine-grained categorization problems. The best performance obtained in this database was reported by Yao [10] and it uses a very specific technique using the template matching approach. We described this technique in Section 2.

6 Conclusion

In this work, we applied families of morphological granulometries to the map of edges of image databases in order to build generic models for some objects. A novel descriptor using this approach is proposed and tested against other widely known descriptors in Caltech-101 database. Our results showed that the proposed descriptor has performance similar to the state-of-the-art descriptors like PHOW, making it a morphological alternative to be used in object recognition problems.

GRABED is used in a novel spatial quantization strategy that is more flexible than the traditional spatial pyramids proposed in [1]. Adjacent regions in a given level may have overlapping areas to avoid breakages in the edge's structures near the border. The number of levels, regions in each level and weights for each level are arbitrary. They are chosen during the validation step in the training.

Finally, we proposed a novel method for fine-grained object categorization that uses models specific for each class. We also tested different types of models using: (1) grouped classes, (2) a multiclass, or (3) a 2-class approach. Models that do not have a confidence larger than thresholds are eliminated. Average probability of all models are used to the final result. Our technique improved the overall results using the BoW approach to solve the fine-grained categorization for this subset of 13 classes of Caltech-UCSD Birds-200 2011 database [12].

Acknowledgments. The authors are partially supported by CNPq, the Brazilian National Research Council.

References

1. Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: CVPR, New York, pp. 2169–2178 (2006)
2. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR, San Francisco, pp. 3360–3367 (2010)
3. Bay, H., Ess, A., Tuytelaars, T., Gool, L.: Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding 110, 346–359 (2008)

4. Selim, S., Ismail, M.: K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. IEEE Trans. on PAMI 6, 81–87 (1984)
5. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proc. ACM Int. Conf. on Image and Video Retrieval, pp. 401–408 (2007)
6. Matheron, G.: Random sets and integral geometry, vol. 261. Wiley, New York (1975)
7. Newell III., J.: Pixel classification by morphological granulometric features. Thesis, Rochester Institute of Technology (1991)
8. Soille, P.: Morphological Image Analysis: Principles and Applications, 2nd edn. Springer (2002)
9. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR, San Diego, USA, pp. 886–893 (2005)
10. Yao, B., Bradski, G., Fei-Fei, L.: A codebook-free and annotation-free approach for fine-grained image categorization. In: CVPR, Providence, USA, pp. 3466–3473 (2012)
11. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. California Institute of Technology (2011) CNS-TR-2011-001
12. Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., Belongie, S.: Visual recognition with humans in the loop. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 438–451. Springer, Heidelberg (2010)
13. Lara, A., Hirata Jr., R.: Combining features to a class-specific model in an instance detection framework. SIBGRAPI (2011)
14. Canny, J.: A computational approach to edge detection. IEEE Trans. on PAMI 8, 679–698 (1986)
15. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. IEEE Trans. on PAMI 28, 1465–1479 (2006)
16. Bosch, A., Zisserman, A., Munoz, X.: Image Classification using Random Forests and Ferns. In: ICCV, Rio de Janeiro, Brazil, pp. 1–8 (2007)
17. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence, pp. 1137–1145 (1995)
18. Fei-Fei, L., Fergus, R., Perona, P.: One-Shot Learning of Object Categories. IEEE Trans. on PAMI 28, 594–611 (2006)
19. Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. In: ICCV, Nice, France, pp. 1470–1478 (2003)
20. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval. Addison-Wesley Longman Publishing Co. Inc., Boston (2011)
21. Chih-Fong, T.: Bag-of-Words Representation in Image Annotation: A Review. ISRN Artificial Intelligence 2010 (2012)
22. Ojala, T., Pietikainen, M., Harwood, D.: Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: 12th IAPR Int. Conf. on Pattern Recognition, pp. 582–585 (1994)
23. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 694–699 (2002)

Qualitative Comparison of Contraction-Based Curve Skeletonization Methods

André Sobiecki[1], Haluk C. Yasan[2], Andrei C. Jalba[2], and Alexandru C. Telea[1]

[1] Institute Johann Bernoulli, University of Groningen, The Netherlands
{a.sobiecki,a.c.telea}@rug.nl
[2] Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
h.c.yasan@student.tue.nl, a.c.jalba@tue.nl

Abstract. In recent years, many new methods have been proposed for extracting curve skeletons of 3D shapes, using a mesh-contraction principle. However, it is still unclear how these methods perform with respect to each other, and with respect to earlier voxel-based skeletonization methods, from the viewpoint of certain quality criteria known from the literature. In this study, we compare six recent contraction-based curve-skeletonization methods that use a mesh representation against six accepted quality criteria, on a set of complex 3D shapes. Our results reveal previously unknown limitations of the compared methods, and link these limitations to algorithmic aspects of the studied methods.

Keywords: Curve skeletons, shape analysis, shape representation.

1 Introduction

Curve skeletons are among the most well-known, and widest used, descriptors for 3D shapes. They have been extensively used in applications such as shape matching and recognition, computer animation, virtual navigation, and shape processing [8,27].

Earlier methods for computing curve skeletons used mainly voxel-based 3D shapes. In recent years, several methods have been proposed to compute curve skeletons from meshed 3D shapes, using a *contraction* principle, where the input mesh is iteratively shrunk towards its local center. Such methods are highly computationally scalable, and can easily handle mesh shapes with considerable more details than voxel-based methods. However, their algorithmic complexity makes it harder to reason analytically about the properties of the produced skeletons. In particular, it is not fully clear how their results relate to desirable skeleton properties. Moreover, since such methods are typically compared with methods in the same class (mesh-based), it is unclear whether mesh-based methods are indeed always superior to voxel-based methods.

In this paper, we compare six mesh-contraction-based curve-skeletonization methods, all which are based on a collapse principle, against six accepted quality criteria: centeredness, homotopy to the input shape, invariance under isometric

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 425–439, 2013.

transformations, detail preservation, smoothness, and independence from the input shape's sampling. Our work extends the earlier survey of Cornea *et al.* [8] by studying six mesh-based curve-skeletonization algorithms published after that survey was done. Our results reveal several limitations of the studied methods which, to our knowledge, have not been highlighted in the literature, and link these to algorithmic aspects of the studied methods.

The structure of this paper is as follows. Section 2 overviews related work in curve skeletonization, with a focus on contraction-based methods. Section 3 details the quality criteria used for the comparison. Section 4 presents the comparison results. Section 5 discusses our findings. Section 6 concludes the paper with future work directions.

2 Related Work

For a shape $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega$, we first define its distance transform $DT_{\partial\Omega} : \mathbb{R}^3 \rightarrow \mathbb{R}^+$

$$DT_{\partial\Omega}(\mathbf{x} \in \Omega) = \min_{\mathbf{y} \in \partial\Omega} \|\mathbf{x} - \mathbf{y}\|. \tag{1}$$

The surface skeleton of Ω is next defined as

$$S(\Omega) = \{\mathbf{x} \in \Omega \,|\, \exists\, \mathbf{f}_1,\, \mathbf{f}_2 \in \partial\Omega, \mathbf{f}_1 \neq \mathbf{f_2}, \|\mathbf{x} - \mathbf{f}_1\| = \|\mathbf{x} - \mathbf{f}_2\| = DT_{\partial\Omega}(\mathbf{x})\} \tag{2}$$

where $\mathbf{f}_1$ and $\mathbf{f}_2$ are the contact points with $\partial\Omega$ of the maximally-inscribed ball in Ω centered at $\mathbf{x}$ [12,24], also called *feature transform* (FT) points [15]. Surface skeletons consist of several manifolds with boundaries which meet along a set of Y-intersection curves [9,17,7]. They can be computed by voxel-based or mesh-based methods [21,5,28,14,3,14]. A recent comparison of surface-skeleton extraction methods is given in [15].

In contrast to surface skeletons, curve skeletons are loosely defined as 1D structures "locally centered" within the input shape Ω. The lack of a unanimously accepted formal definition has led to many methods which compute curve skeletons following not necessarily identical definitions. This makes it hard to analytically compare, and reason about, the properties of the produced curve skeletons.

Tools from mathematical morphology [25] were among the first used to compute curve skeletons: The residue of openings, based on Lantuéjoul's formula [16], usually leads to disconnected skeleton branches, whereas methods based on homotopic thinning transformations [16,19,4,21] yield connected skeletons. Combining such techniques with distance-driven thinning further ensures the extraction of centered surface and curve skeletons [1].

Dey and Sun propose one of the first analytic definitions of curve skeletons based on the medial geodesic function (MGF), where the curve skeleton is defined as the locus of points having at least two equal-length shortest geodesics on $\partial\Omega$ between their feature points [10,23]. Reniers *et al.* extend the MGF to regularize curve skeletons by assigning each skeleton point an importance equal to the area bounded by such geodesics, in a voxel setting [24], inspired by the so-called 2D collapse metric [20,32]. A GPU implementation of the above metric for mesh models is presented in [15].

Voxel-based methods typically require significant resources to store and process the large voxel volumes required to capture the fine details of complex 3D shapes. To be used on 3D meshes, such methods require a costly voxelization step. Mesh-based methods address these cost issues by working directly on a mesh representation of $\partial\Omega$. In recent years, several such mesh-based methods have been proposed based on a *contraction* principle, which shrinks the input mesh until the 1D curve-skeleton structure is reached, as follows. Au *et al.* shrink the mesh via Laplacian smoothing until its volume gets close to zero, followed by an edge-collapse (to extract the 1D curve skeleton) and a re-centering step (to correct shrinking errors) [2]. Cao *et al.* extend this idea to extract curve skeletons from incomplete point clouds [6]. The ROSA method defines, and extracts, curve skeletons using rotational, rather than positional, symmetry: $\partial\Omega$ is cut with planes, and curve-skeleton points are found as the centers of planes which minimize the variance between the plane's normal and $\partial\Omega$ normals along the cut curve [30]. Sharf *et al.* reverse the contraction direction: They find the curve skeleton as the centers of a set of competing fronts which evolve to approximate the input surface [26]. A similar method is presented by Hassouna and Farag [13]. Telea and Jalba define, and extract, curve-skeletons by contracting the surface skeleton $S(\Omega)$ (computed as in [18]) inwards, along the gradient of the 2D distance transform of $\partial S(\Omega)$, *i.e.* define the curve-skeleton as the result of a two-step skeletonization [31].

Mesh-contraction methods are currently deemed to be the state-of-the-art for extracting detailed curve skeletons from high-resolution shapes [29]. As 3D models become more complex, it is arguable that such methods will dominate the more costly voxel-based methods. Conceptually, such methods work very similarly to voxel-based thinning. However, there are few, if any, comparisons of contraction-based methods based on the accepted skeleton desirable criteria used for earlier voxel-based methods. Also, the algorithmic complexity of mesh-contraction methods makes a formal analysis thereof more complex than for voxel-based methods. All in all, it is not clear if mesh-contraction methods are indeed always superior to voxel-based methods, and if not, which are their specific weak points with respect to desirable skeleton criteria.

3 Comparison Criteria

The literature knows a well-accepted set of quality criteria that curve skeletons should conform to. For curve-skeletonization methods, such criteria are significantly more important than for surface skeletonization methods: While the latter can be rigorously checked against the formal surface skeleton definition (Eqn. 2), the former do not use a single curve-skeleton definition. As such, the only comparison available for curve skeletons is a qualitative one, from the perspective of desirable quality criteria. Following [8,15,27], we focus on the following generally-accepted quality criteria for a curve skeleton:

Homotopy: The curve skeleton is topologically-equivalent with the input shape, *i.e.* has the same number of connected components and tunnels.

Invariant: The curve skeleton should be invariant under isometric transformations of the input shape.

Thin: The curve skeleton should be as thin as the sampling model used allows it. Voxel-based curve skeletons should be one voxel thick. Mesh-based curve skeletons should contain only lines, and not polygons or loose points. Point-cloud based curve skeletons should ideally have zero local thickness in any direction orthogonal to the largest eigenvector of the covariance matrix of point neighborhoods.

Centered: This is the hardest criterion to quantify, since it is not uniquely defined when a curve is centered within a 3D shape. However, several weak forms of curve-skeleton centeredness exist: The curve skeleton should be a subset of the surface skeleton (since the latter is by definition centered within the shape); and in no case should the curve-skeleton exit the input shape.

Smoothness: As centeredness, smoothness is also hard to formally define. Surface skeleton manifolds are known to be at least $\mathcal{C}^2$ continuous [22,27]. Curve-skeletons are centered subsets thereof [29,31]. Hence, it is arguable that curve skeletons should be also piecewise, *i.e.* per branch, $\mathcal{C}^2$. In any case, curve skeletons should not exhibit curvature discontinuities induced by the sampling of either the input surface or curve skeleton representation.

Detail Preserving: Curve skeletons should be able to capture fine-scale details, such as bumps, of the input shape, in a user-controlled manner. In other words, the user should be able to select the scale of input shape details which the curve skeleton should capture (being significant) and the scale of details to ignore (being regarded as noise).

Sampling Robustness: Given two different samplings of an input shape (*e.g.* two different level-of-detail meshes), the difference between the two corresponding curve skeletons should be proportional with the difference of the two input meshes. In other words, small input-sampling differences should not cause large differences in the curve skeleton.

4 Comparison

Given our core question on how curve-skeletonization methods perform, we compared six such methods (further denoted in the paper by the abbreviations listed below):

Au *et al.* (AU) [2]: We included this method as it is arguably the best-known mesh-based skeletonization technique in existence [13,29,15].

Tagliasacchi *et al.* (ROSA) [30]: We chose this method given its advocated noise-resistance and since it works on point clouds, which is a different type of input than the other methods.

Cao *et al.* (CAO) [6]: We chose this method since it uses a contraction similar to [2], but works on point clouds, like [30].

Telea and Jalba (TJ) [31]: In contrast to all other curve-skeletonization methods, this technique contracts the surface skeleton, rather than the input mesh, to compute the curve skeleton. It produces a point cloud rather than a polyline curve-skeleton. For comparison fairness, we postprocessed the produced point cloud using the polyline reconstruction proposed in [2].

We also developed and tested two extensions of [2], as follows.

Au *et al.* Improved (AUI): A well-known limitation of Au *et al.* is its skeleton re-centering step [29]. As the input mesh is contracted, it can go off-center due to numerical and discretization inaccuracies of the Laplacian smoothing. To address this issue, we proceed as follows. During the Laplacian contraction and edge-collapse steps of the method, we maintain a backwards, skeleton-node-to-mesh-vertex mapping $\Pi : S \to \partial\Omega$, which can be used to identify those mesh vertices $\mathbf{v} \in \partial\Omega$ that 'collapsed' into a given skeleton node $\mathbf{s} \in S(\Omega)$. The re-centering step uses Π to compute the final position of each node $\mathbf{s}$ as a weighted average of the vertices in $\Pi(\mathbf{s})$, with weights given by the areas of the input-mesh triangles with vertices in $\Pi(\mathbf{s})$.

Au *et al.* Using Surface Skeletons (AUS): The improved re-centering outlined above cannot fully correct errors accumulated during the iterative contraction. To further reduce these, we start the Laplacian contraction from the surface skeleton, which is closer to the final target (curve skeleton) than the input mesh, along the idea proposed in [31].

Global Considerations: In our method choice, we focused on recent contraction-based techniques, not studied in the survey of Cornea *et al.* [8], proven by their authors on complex shapes, and which use different curve-skeleton detection principles. All studied methods satisfy the *invariance* criterion by construction, since they work in 3D vector space. All methods also directly satisfy the *thinness* criterion, since they model the curve-skeleton as a polyline. We used the original implementations provided by their authors, all running on a Windows PC with 4 GB RAM. Since not all studied methods claim computational efficiency, we excluded timings from the comparison.

Comparison Material: For comparison, we used a set of 21 3D shapes which are frequently encountered in the curve-skeleton literature (for details, see [34]). Figures 1, 2, 3 and 4 and show relevant samples from this set, within space limitations. The models have between 20K and 300K vertices. We used MeshLab [33] to clean mesh models for normal orientation consistency, T-vertices, and duplicate vertices. To factor our parameter settings, we ran each method for uniformly-sampled values of all its documented parameters, and retained in our final comparisons the best results with respect to the quality criteria mentioned in Sec. 3.

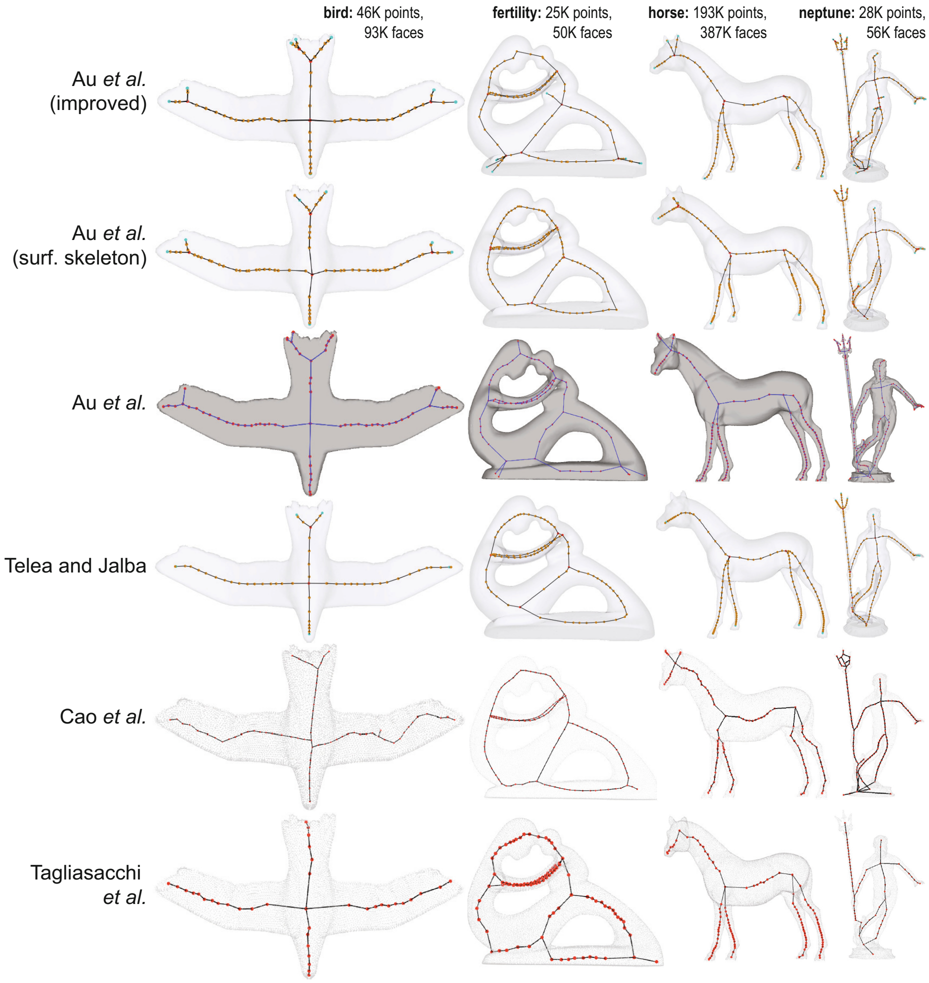

bird: 46K points, 93K faces
fertility: 25K points, 50K faces
horse: 193K points, 387K faces
neptune: 28K points, 56K faces
Au et al. (improved)
Au et al. (surf. skeleton)
Au et al.
Telea and Jalba
Cao et al.
Tagliasacchi et al.

Fig. 1. Overview comparison of skeletonization methods

4.1 Overview

Figure 1 shows an overview of several curve skeletons extracted by the compared methods. Even at this level, we quickly notice that not all skeletons are equally well centered, equally smooth, and have the same number of terminal (detail) branches. We next zoom-in on each criterion and discuss our findings with respect to the studied methods.

4.2 Homotopy

For relatively simple shapes of genus 0 or higher, all studied methods behaved equally well, *i.e.* produced curve skeletons homotopic with the input shape (Fig. 1). This is due to the fact that all these methods start by contracting the input shape *and* change only the geometry, but not the topology, of this shape during the iterative contraction process. Still, detail differences exist between these methods. Skeleton junctions are not always identical, so the produced skeleton graph is different, see *e.g.* the marked limbs-to-body junctions of the *bird* model in Fig. 2 (left) and the *horse* model in Fig. 3 (right). Differences get larger for small-scale details, where curve skeleton terminal branches enter saliencies of the input shape, see *e.g.* Fig. 3 (*neptune*, *frog*). An extreme case happens when the input mesh has self-intersections, *e.g.* Fig. 2 (*frog*). Here, CAO and ROSA create curve skeletons whose topology is far from the input shape (fake loops and branches).

4.3 Centeredness

The methods AU, AUI, and AUS produce similar, well centered, results. Among these, AUS is the best: Since contraction starts from the surface skeleton, nodes go less off-center, as the surface skeleton is already centered by definition and closer to the curve skeleton than the input mesh. For mesh-based methods, TJ produced the best centering. This is due to the fact that TJ contracts the surface skeleton along the gradient field of its 2D distance transform, which is by definition tangent to the surface skeleton itself, so the curve skeleton stays inside the surface skeleton by construction. In contrast, AU, AUI, and AUS contract in the direction of the shrunken surface's normals. These are delicate to estimate as the shape shrinks and develops singularities (creases). The different re-centering steps performed by these methods alleviate, but cannot fully correct, these problems.

ROSA's results are quite poorly centered in several areas. As mentioned in [30], orientation information is unreliable around junctions, where the input shape has many points with diverse orientations. To overcome this, ROSA treats junctions specially. This works well for junctions whose branches correspond to tubular shape parts of similar size. However, we discovered that junctions where shape parts of very different sizes and shapes meet create problems, see *e.g.* Fig. 2 for the *bird* model (wings joining rump) and *neptune* (arm-torso junction).

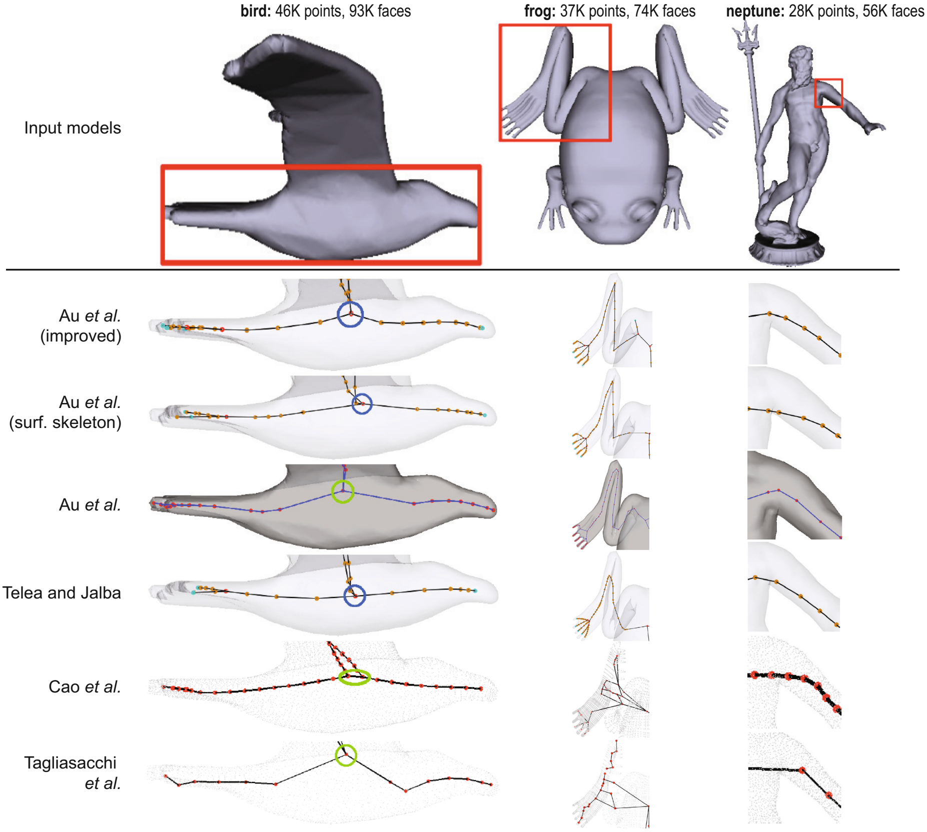

bird: 46K points, 93K faces
frog: 37K points, 74K faces
neptune: 28K points, 56K faces
Input models
Au et al. (improved)
Au et al. (surf. skeleton)
Au et al.
Telea and Jalba
Cao et al.
Tagliasacchi et al.

Fig. 2. Centeredness comparison. Details show areas marked by red insets.

The *frog* model (Fig. 2) reveals two other challenges. First, the model has several very sharp bends around the leg joint. Secondly, in the same area, the mesh has several self-intersections. Meshless methods (CAO, ROSA) generate seriously erroneous skeletons here, and even skeleton disconnections. In these areas, TJ still creates a smooth skeleton, but cannot handle centeredness perfectly. This is due to the fact that the surface skeleton it starts from has errors in self-intersecting areas, since the technique used to compute it [15] cannot handle self-intersecting surfaces. In contrast, AU, AUI, and AUS generate very similar, relatively well-centered, skeletons in these challenging areas.

The *neptune* model (Fig. 2) highlights the situation where a relatively thin object part (arm) joins a thick one (torso). In such areas, curve (and surface) skeletons exhibit so-called ligature branches which connect the skeleton branches of the two parts [22]. If the two parts form an angle different from 90°, like in our case, the ligature branch has to rapidly turn [27]. This turn is best captured by AU. In contrast, all other methods emphasize smoothness too much, which results in clearly off-centered skeletons close to the armpit.

4.4 Detail Preservation

Detail preservation refers to the generation of separate curve-skeleton terminal branches for all input shape bumps, or salient convexities, at a user-specified scale. Detail preservation is important for applications such as shape matching, retrieval, and reconstruction [8,24]. Large details, such as the limbs of shapes in Fig. 1, are well captured by skeleton branches by all studied methods. For smaller-scale details, the situation is different, see Fig. 3 left. The problem is that all studied methods include explicit actions to smooth the computed skeletons. Although desirable (see next Sec. 4.5), such smoothing will remove some small-scale branches.

AU and AUI preserve small-scale, detail, branches best. In contrast, AUS and TJ find detail branches of long protrusions (*e.g.* Fig. 3, *neptune* and *frog* fingers) quite well, but fail to find branches for shallower bumps, such as *gargoyle*'s wing-tips. Upon closer analysis, we found that this is caused by the fact that the surface skeletons that both AUS and TJ start from, fail to capture such details. Hence, these details cannot appear further in the curve skeleton.

CAO and ROSA perform the worst for this criterium. These methods fail finding most detail skeleton branches found by the other studied methods. Moreover, when found, small-scale terminal skeleton branches seem to be arbitrary, as Fig. 3 shows for all three models on the left.

Small-scale noise is ignored equally well by all methods. For all the studies mesh-based methods, this is an effect to their built-in smoothing, which appears to work well at small scale.

4.5 Smoothness

As outlined earlier, curve-skeleton branches should be at least $\mathcal{C}^2$ continuous curves (Sec. 3). Hence, skeletonization methods should follow this property as well as possible. Voxel methods are inherently constrained here by the sampling resolution. In contrast, mesh-based methods which model the curve skeleton as a polyline should distribute the computed skeletal points, or sample the skeleton, to optimally approximate the desired smooth curve. Hence, for these methods, the issue of skeleton smoothness is implicitly connected to the skeletal curve sampling.

Contraction-based methods, as the ones we studied, have an additional challenge here. As the input mesh is contracted, the local point density naturally increases in convex areas and decreases in concave ones. This potentially leaves too few nodes to approximate well the curve skeleton in concave areas. Ligature branches are an extreme case hereof. An example are the ligature branches that connect the horse's leg-skeletons to its rump-skeleton (Fig. 3 right). Here, CAO, ROSA, and up to some extent AU, clearly show a lower point density – see branches meeting at the marked junctions. This in turn creates spurious kinks in the rump's curve skeleton. In contrast, AUS, AUI, and TJ create smoother skeletons. The skeletons of TJ and AUS follow the rump's curvature best. This is explained by the fact that their contraction is constrained to stay on

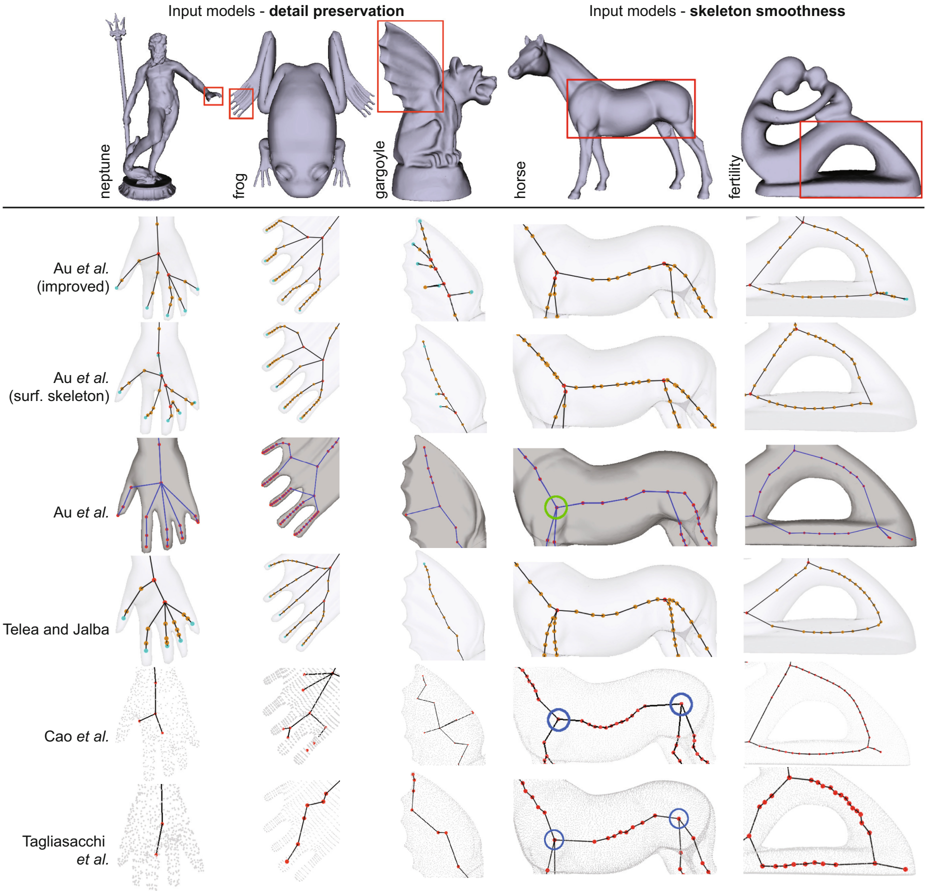


Fig. 3. Comparison for detail preservation comparison (left) and skeleton smoothness (right)

the surface skeleton, whose shape already captures the input shape's curvature. AU and AUI both fail capturing the rump's curvature, since they have no such constraint. The same non-uniform skeletal point distribution is also observed for the *fertility* model (Fig. 3 right). Here, again, AUS and AUI yield the most uniform point distribution, and ROSA and AU the least uniform one (which leads to unnatural kinks).

4.6 Sampling Robustness

Sampling robustness refers to the relation between the resolution of the input shape and changes in its curve skeleton. Ideally, we would like that when the former changes slightly, the curve skeleton also changes only slightly. This property is closely related to the concept of *regularization*, which states that small changes in the input shape Ω should only yield small changes in its skeleton [32,24,15].

To study this, we produced three versions of the *dragon* model (see Fig. 4), using the Yams mesh resampling tool [11]. Next, we ran the studied skeletonization methods on these datasets, and analyzed the results. In the comparison, we had to exclude CAO and ROSA, as the provided implementations of these methods were too slow to complete, even in several hours, for the largest-resolution meshes.

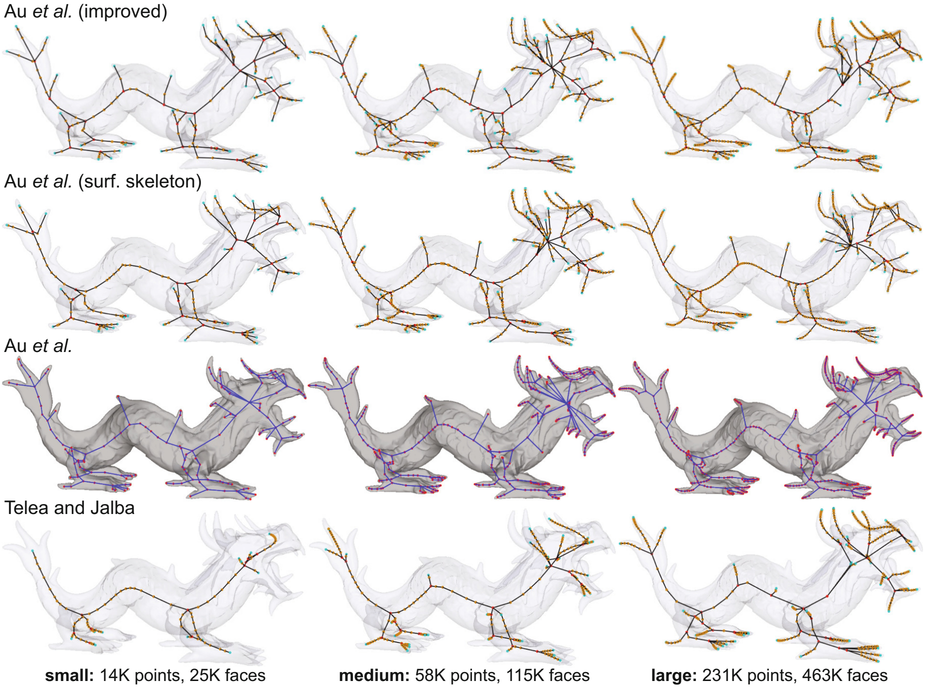

Au *et al.* (improved)
Au *et al.* (surf. skeleton)
Au *et al.*
Telea and Jalba
small: 14K points, 25K faces
medium: 58K points, 115K faces
large: 231K points, 463K faces

Fig. 4. Sampling robustness comparison

The method AU is quite sensitive to the mesh sampling. Looking at Fig. 4, we see that, in the dragon head area, the small and large resolution models produce relatively similar skeletons, but the medium-resolution model yields a very different skeleton topology. Given that higher resolution can only potentially add extra details, but not remove existing ones, we expect to get an increasingly rich curve skeleton (in terms of terminal branches), but the core structure of this skeleton should not change significantly. This is not the case, which hints to an important instability of the method with respect to mesh resolution.

In contrast, AUS and AUI show a much stabler curve skeleton with respect to mesh resolution. Although these methods do not produce identical skeletons for the same resolution, the changes of their respective skeletons as the resolution changes, are quite small. Both methods find more terminal skeleton branches as the resolution increases, which is expected since higher-resolution models capture more surface details.

The TJ method is the most sensitive to sampling. For the low-resolution model, the method simply fails to extract many significant branches. Although more branches are found for the high resolution model, many significant surface details, like the upper spikes on the back and tail, fail to generate branches. This can be directly traced to the quality of the surface skeleton: The underlying method used to compute it [18] produces as many skeleton points as surface points. To accurately capture the surface skeletal structure, very densely-sampled models are required [15]. Less densely sampled surface skeletons will in turn create a noisy distance-transform gradient, which will contract the skeleton mesh in the wrong directions.

5 Discussion

Contrary to our initial belief, based on the studied contraction-based skeletonization literature, all contraction methods studied here appeared to be much more sensitive in terms of *all* studied quality criteria (except homotopy) than implied by the examples in the literature. The CAO and ROSA methods performed significantly under expectations. The AU method performed relatively well for smooth shapes, but showed limitations for centeredness and smoothness for more complex shapes. This is the main reason for us having designed the two improved variants AUI and AUS. The trade-off between these variants is as follows: While AUS yields smoother skeletons, AUI delivers a better centeredness. The TJ method dominates all others in terms of smoothness, but has clear centeredness problems in ligature areas, and requires a very high input mesh sampling to generate even moderately-detail skeleton branches, due to its usage of the surface skeleton.

A key question is whether voxel-based skeletonization methods can overcome the above limitations. Although answering this deserves a separate study, we outline below several observations in this respect:

Homotopy: Voxel-based skeletonization methods are not, by definition, homotopy preserving. For example, Reniers *et al.* can occasionally create small disconnected components [24]. However, thinning methods can enforce homotopy relatively easily [21,1].

Invariance: Like for homotopy, voxel methods are not invariant under isometric transformations by construction, as mesh-based methods are. Using truly Euclidean distance transforms helps invariance [14], but does not guarantee it [27].

Thinness: For voxel methods, this criterion translates to creating one-voxel-thin skeletal manifolds and curve skeletons. Thinning methods are best suited to enforce thinness [21,1], whereas general-field methods cannot guarantee it [14,24].

Applying a thinning postprocessing step (in line with the former methods) to general-field methods appears to be the optimal combination.

Centeredness: Just as thinness, centeredness for voxel methods is limited by sampling resolution. Using an Euclidean distance transform to drive thinning [1] or, alternatively, to find the skeleton by directly applying Eqn. 2[10,24], guarantees thinness under this sampling limitation. In contrast, all contraction methods studied here do not use a volumetric distance transform. Hence, their centeredness is subject to accumulated errors during the iterative contraction process, which arguably makes them less accurate than voxel methods.

Detail preservation: This requirement is not fundamentally linked to the type of object discretization (mesh or voxel-based). Using a global importance metric can guarantee detail preservation for both mesh [15], voxel [24], or hybrid [10] representations.

Smoothness: Voxel models can be preferable to mesh contraction models here. Indeed, while the former typically treat each skeleton voxel separately, the latter enforce local constraints on the surface and curve skeletons. This forces mesh-based skeletons to be either smooth (but not well centered), or well centered (but not smooth).

An additional desirable property of skeletons is *reconstructability*, *i.e.*, the ability to reconstruct the input shape from its skeleton. Surface skeletons should obey this property, by definition, as the medial axis transform (MAT) is a dual of the input shape [27]. Curve skeletons can obey this property only partially. Although reconstructability is sometimes studied as a separate property [8,1], it can be traced directly to a combination of centeredness and detail preservation.

The main challenge we find for voxel methods is, however, scalability: Voxelizing complex meshes to resolutions over 1000^3 voxels, and further processing such volumes to extract curve or surface skeletons, is much slower, and more memory demanding, than using mesh-based methods. For instance, a highly optimized parallel implementation of [24] processes the 700^3 *dragon* model (Fig. 4) in around 15 minutes; the equivalent mesh model (463K faces) is processed in under a minute by all studied mesh-based methods. Moreover, the memory consumption of voxel methods is at least an order of magnitude larger than for mesh-based methods. If efficient data representation and GPU parallelization schemes were designed to reduce this overhead, voxel-based methods may in the end be a very strong competitor to mesh-based methods.

6 Conclusions

In this paper, we have presented a qualitative comparison of six contraction-based curve-skeletonization methods that use a mesh representation of the input shape to be skeletonized. The methods were compared from the perspective of several accepted quality criteria: homotopy, thinness, centeredness, detail preservation, smoothness, and robustness to sampling. In contrast to recent insights from the mesh skeletonization literature, the studied mesh-based methods appeared to perform less optimal than expected.

Although our comparison is far from exhaustive, it raises a number of important points about the current state of mesh-based curve skeletonization techniques. First and foremost, the question is raised whether such methods can outclass earlier voxel-based skeletonization methods (if we ignore computational resources). A more critical more critical quantitative and qualitative evaluation of such algorithms against each other and also against voxel-based skeletonization methods is needed to answer this question. Finally, we believe that our comparison will generate increased attention towards the development of efficient algorithms that exploit the desirable properties of voxel-based skeletonization techniques.

References

1. Arcelli, C., Sanniti di Baja, G., Serino, L.: Distance-driven skeletonization in voxel images. IEEE TPAMI 33(4), 709–720 (2011)
2. Au, O., Tai, C., Chu, H., Cohen-Or, D., Lee, T.: Skeleton extraction by mesh contraction. Proc. ACM SIGGRAPH, 44:1–44:10 (2008)
3. Bai, X., Latecki, L.J., Liu, W.-Y.: Skeleton pruning by contour partitioning with discrete curve evolution. IEEE TPAMI 3(29), 449–462 (2007)
4. Beucher, S.: Digital skeletons in Euclidean and geodesic spaces. Signal Processing 38(1), 127–141 (1994)
5. Bouix, S., Siddiqi, K., Tannenbaum, A.: Flux driven automatic centerline extraction. Med. Imag. Anal. 9(3), 209–221 (2005)
6. Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., Su, Z.: Point cloud skeletons via laplacian based contraction. In: Proc. SMI, pp. 187–197 (2010)
7. Chang, M., Leymarie, F., Kimia, B.: Surface reconstruction from point clouds by transforming the medial scaffold. CVIU (113), 1130–1146 (2009)
8. Cornea, N., Silver, D., Min, P.: Curve-skeleton properties, applications, and algorithms. IEEE TVCG 13(3), 530–548 (2006)
9. Damon, J.: Global medial structure of regions in $\mathbf{R}^3$. Geometry and Topology 10, 2385–2429 (2006)
10. Dey, T., Sun, J.: Defining and computing curve-skeletons with medial geodesic function. In: Proc. SGP, pp. 143–152 (2006)
11. Frey, P.: YAMS: a fully automatic adaptive isotropic surface remeshing procedure. tech. rep. 0252, INRIA (November 2001), http://www.ann.jussieu.fr/frey
12. Giblin, P., Kimia, B.: A formal classification of 3D medial axis points and their local geometry. IEEE TPAMI 26(2), 238–251 (2004)
13. Hassouna, M., Farag, A.: Variational curve skeletons using gradient vector flow. IEEE TPAMI 31(12), 2257–2274 (2009)
14. Hesselink, W., Roerdink, J.: Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. IEEE TPAMI 30(12), 2204–2217 (2008)
15. Jalba, A., Kustra, J., Telea, A.: Surface and curve skeletonization of large meshes on the GPU. IEEE TPAMI (2012), doi:10.1109/TPAMI.2012.212

16. Lantuéjoul, C.: La squelettisation et son application aux mesures topologiques de mosaiques polycristallines. PhD thesis, School of Mines, Paris (1979)
17. Leymarie, F., Kimia, B.: The medial scaffold of 3D unorganized point clouds. IEEE TVCG 29(2), 313–330 (2007)
18. Ma, J., Bae, S., Choi, S.: 3D medial axis point approximation using nearest neighbors and the normal field. Vis. Comput. 28(1), 7–19 (2012)
19. Meyer, F.: Skeletons and perceptual graphs. Signal Processing 16(4), 335–363 (1989)
20. Ogniewicz, R.L., Kubler, O.: Hierarchic Voronoi skeletons. Patt. Recog. (28), 343–359 (1995)
21. Palágyi, K., Kuba, A.: Directional 3D thinning using 8 subiterations. In: Bertrand, G., Couprie, M., Perroton, L. (eds.) DGCI 1999. LNCS, vol. 1568, pp. 325–336. Springer, Heidelberg (1999)
22. Pizer, S., Siddiqi, K., Szekely, G., Damon, J., Zucker, S.: Multiscale medial loci and their properties. IJCV 55(2-3), 155–179 (2003)
23. Prohaska, S., Hege, H.C.: Fast visualization of plane-like structures in voxel data. In: Proc. IEEE Visualization, pp. 29–36 (2002)
24. Reniers, D., van Wijk, J.J., Telea, A.: Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. IEEE TVCG 14(2), 355–368 (2008)
25. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, London (1982)
26. Sharf, A., Lewiner, T., Shamir, A., Kobbelt, L.: On-the-fly curve-skeleton computation for 3d shapes. CGF 26(3), 323–328 (2007)
27. Siddiqi, K., Pizer, S.: Medial Representations: Mathematics, Algorithms and Applications. Springer (2009)
28. Sud, A., Foskey, M., Manocha, D.: Homotopy-preserving medial axis simplification. In: Proc. SPM, pp. 103–110 (2005)
29. Tagliasacchi, A., Alhashim, I., Olson, M., Zhang, H.: Mean curvature skeletons. CGF 31(5), 1735–1744 (2012)
30. Tagliasacchi, A., Zhang, H., Cohen-Or, D.: Curve skeleton extraction from incomplete point clouds. ACM TOG 28(3), 71:1–71:9 (2009)
31. Telea, A., Jalba, A.: Computing curve skeletons from medial surfaces of 3D shapes. In: Proc. TPCG, pp. 137–145 (2012)
32. Telea, A., van Wijk, J.J.: An augmented fast marching method for computing skeletons and centerlines. In: Proc. VisSym, pp. 251–259 (2002)
33. Univ. of Pisa. MeshLab mesh processing toolkit (2012), `http://meshlab.sourceforge.net`
34. Yasan, H.: Contraction-based curve skeletonization of 3D meshes, MSc thesis, Dept. of Computer Science, TU Eindhoven, the Netherlands (2012), `http://repository.tue.nl/741246`

Detection of Texture and Isolated Features Using Alternating Morphological Filters

Igor Zingman[1], Dietmar Saupe[1], and Karsten Lambers[2]

[1] Department of Computer and Information Science,
University of Konstanz, Germany
[2] Institute of Archaeology, Heritage Sciences and Art History,
University of Bamberg, Germany

Abstract. Recently, we introduced a morphological texture contrast (MTC) operator that allows detection of textural and non-texture regions in images. In this paper we provide comparison of the MTC with other available techniques. We show that, in contrast to other approaches, the MTC discriminates between texture details and isolated features, and does not extend borders of texture regions. Using the ideas underlying the MTC operator, we develop a complementary operator called morphological feature contrast (MFC) that allows extraction of isolated features while not being confused by texture details. We illustrate an application of the MFC operator for extraction of isolated objects such as individual trees or buildings that should be distinguished from forests or urban centers. We furthermore provide an example of how this operator can be used for detection of isolated linear structures. We also derive an extended version of the MFC that works with vector-valued images.

1 Introduction

This paper focuses on the problem of distinguishing isolated features, also called individual features, from features that are part of a texture[1]. We refer to the latter features as texture details. This problem may occur when one wants to detect texture regions, but distinguish them from isolated features that should not be assigned to a texture class. The dual problem occurs when it is necessary to detect isolated features avoiding detection of parts of neighboring or background texture even if texture details are similar to features of interest. Here we consider both problems, namely detection of texture and of individual features.

Although a large variety of texture classification methods has been developed, much less attention has been given to the apparently simpler problem of texture detection that discriminates between high contrast texture (of any type) and non-texture regions. This is not a simple task if accurate localization is required and if texture must be distinguished from individual features.

In [1] it was proposed to use the difference between maximal and minimal intensities (MaxMin difference) in a pixel neighborhood for a fast segmentation

[1] By features we mean small image elements, for example blobs, ridges or edges.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 440–451, 2013.

of an image into textured and non-texture regions. A standard deviation (StD) is frequently used as a measure of texture that describes its smoothness [2]. In [3], where the Local Binary Patterns (LBP) approach was developed, the authors also suggested to incorporate a variance-based descriptor for texture classification purposes. While the LBP descriptor is related to inherent texture properties, a complementary variance-based descriptor measures texture contrast. The amplitude modulation function (AMF), derived from the amplitude-modulation frequency-modulation model [4], can locally capture texture contrast. Though each of the texture contrast descriptors mentioned above can be used to discriminate between texture regions and non-texture areas, also called smooth areas in this paper, they cannot distinguish individual features from texture details.

Recently, several descriptors were suggested to approach this problem. In [5] the PanTex index was developed to detect settlements in panchromatic satellite imagery. The operator is able to distinguish texture areas from individual linear features such as roads or borders between homogenous cultivated fields in satellite images. The PanTex index is defined as a minimal contrast among contrast measures derived from the gray-level co-occurrence matrixes (GLCM) [6], computed for different orientations of displacement vectors. The PanTex method, however, does not distinguish other individual features, such as isolated peaks or small isotropic blobs, from texture.

The component count (CC) method [7] is based on the product of two measures computed in small image blocks. The first one is the sum of the number of connected components (component count) in the background and the foreground obtained by simple binarization of image blocks. The second measure is the difference between average intensities in the background and the foreground. This descriptor is supposed to discriminate blocks covering texture and individual step edges at the borders between homogenous regions. A similar idea of counting the number of local extrema (texture primitives) for detection of texture regions was proposed earlier in [8]. Since this method does not take into account contrast of texture primitives, it can be very sensitive to noise.

Another disadvantage that all the above texture descriptors have in common, is that they extend or blur the edges of texture regions preventing accurate localization of texture borders. Recently, we introduced a morphological texture contrast (MTC) descriptor that does not suffer from the above disadvantages [9]. This operator, briefly reviewed in Sec. 2, measures the difference between upper and lower texture envelopes estimated by means of alternating morphological filters [10]. Its qualitative performance was illustrated in [9] using only few remotely sensed images and no quantitative comparison was provided. In Sec. 3 we provide a quantitative comparison using artificially created images and qualitative comparison using a set of standard test images.

As we stated in the beginning of this section, the dual problem to the problem of texture detection is detection of individual features while distinguishing them from texture details. This problem has mainly been treated in the context of edge detection capable of discarding texture surroundings. For example, recently in [11] a surround inhibition mechanism was introduced to improve edge detection at region boundaries. In [12] a normal complexity measure was proposed that is able to separate isolated curves and isolated edges from texture in binary images. The paper provides an original theoretical framework, but it seems to be computationally very expensive.

In Sec. 4 we show how the ideas underlying the MTC operator lead to a Morphological Feature Contrast (MFC) operator that aims at detection of small isolated objects, rather than edges, in textured background. We illustrate the potential of the MFC operator on gray-scale images and also derive its extension to vector-valued images. Additionally, we show how the MFC operator can be incorporated into a scheme for extraction of isolated linear structures.

2 Detection of Texture Regions: The Morphological Texture Contrast (MTC) Operator

Below, we define the morphological texture contrast (MTC) transformation $\psi_{\mathrm{MTC}}(f)$ that we recently introduced in [9] for localization of high contrast textured regions. It uses alternating morphological filters, $\gamma_r\varphi_r$ and $\varphi_r\gamma_r$, which are closing φ followed by opening γ and opening followed by closing, respectively. r denotes the size of the structuring element (SE). Alternating filters are usually employed for noise filtering. We use them to estimate texture envelopes. The difference between upper and lower envelopes[2] defines a measure of texture contrast, which can serve as an indicator of the presence of texture

$$\psi_{\mathrm{MTC}}(f) = |\gamma_r\varphi_r(f) - \varphi_r\gamma_r(f)|^+ \ , \tag{1}$$

where the argument f denotes a 1D signal or a 2D gray-scale image, and $|\cdot|^+$ is defined as

$$|\nu|^+ \triangleq \begin{cases} \nu, & \nu > 0 \\ 0, & \text{otherwise} \ . \end{cases}$$

A remarkable property of these envelopes is that they coincide at individual features, thereby yielding low response at individual features even if they are of high contrast.

The results of applying this transformation to an artificial 1D signal and to a remotely sensed image of a forested area are shown in Fig. 1 and Fig. 2(a,b), respectively. Throughout this paper we use a square SE, where the size refers to its side length. The size r of the SE in Eq. (1) should be chosen to be larger than the maximal distance between details in textured regions. Features that stand apart from texture details farther than r are treated as individual features and are suppressed correspondingly. In general, we can use different sizes $r_1 \neq r_2$ for SEs in $|\gamma_{r_2}\varphi_{r_1}(f) - \varphi_{r_2}\gamma_{r_1}(f)|^+$. While r_1 should be chosen to be larger than the maximal distance between texture details, r_2 should be chosen to be smaller

[2] Since in the 2D case, $\varphi_r\gamma_r$ and $\gamma_r\varphi_r$ are not ordered [13], a lower envelope might be above an upper envelope. However, it can be shown that regions where this happens are small in the sense that an erosion with a structuring element of size r completely removes these regions.

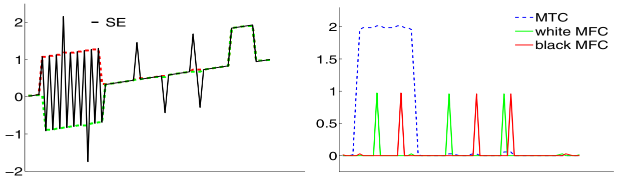
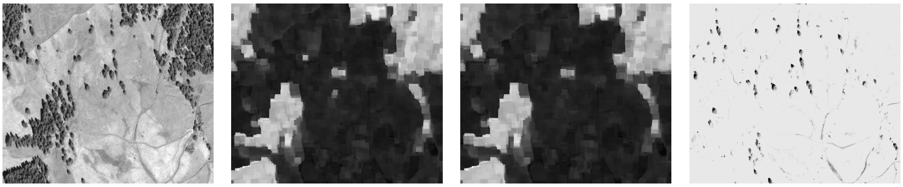


Fig. 1. Left: An artificial signal composed of a slow varying component, a texture region, and individual features. Upper and lower envelopes of the texture obtained with alternating morphological filters are shown by red and green dashed lines. Right: Extraction of the texture region and individual features with the MTC (Sec. 2) and the MFC (Sec. 4) operators.

than the minimal size of texture regions. Thus, using $r_1 \neq r_2$ adds an additional degree of control of the minimal number of clustered details to be regarded as a texture. Comparison of Fig. 2(b) and (c) illustrates this point.



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

Fig. 2. (a) Pan-chromatic satellite image of 1360x1160 pixel size[4]. (b) and (c) The MTC descriptor. $r_1 = r_2 = 30$ in (b) and $r_1 = 30$, $r_2 = 35$ in (c). (d) Extraction of individual dark features, i.e. individual trees, using the ψ^-_{MFC} operator[5] with $r_1 = r_2 = 30$ (Sec. 4). Note that trees in forest areas are almost completely suppressed.

The MTC operator is bias invariant, $\psi_{\mathrm{MTC}}(f) = \psi_{\mathrm{MTC}}(f + a)$, where $a \in \mathbb{R}$ is a constant, invariant to signal inversion[3], $\psi_{\mathrm{MTC}}(f) = \psi_{\mathrm{MTC}}(a - f)$, and proportional to signal magnitude $\psi_{\mathrm{MTC}}(af) = a\psi_{\mathrm{MTC}}(f)$. An important property of the MTC transformation is that it neither extends nor blurs the edges of textured regions, thereby allowing accurate localization of texture borders. This property is illustrated in Fig. 3 in the rightmost column.

3 Comparison of Texture Contrast Descriptors

In this section we qualitatively and quantitatively compare the performance of the MTC algorithm with the CC, the MaxMin difference, the StD, the LBP

[3] Bias invariant and self-complementary operators [14] are invariant to signal inversion.
[4] The image was captured by the GeoEye-1 satellite (© GeoEye 2011).
[5] High values of the transformation are represented by dark tones.

contrast, the AMF, and the PanTex algorithms. We denote by w a scale parameter required for all algorithms. To allow a consistent comparison, a few algorithms were slightly modified as follows.

In the CC algorithm we avoided several parameters suggested by the authors since they need to be adjusted for each type of image. Specifically, we used the simple product of contrast and number of connected components. Instead of disjoint blocks, a sliding window of the single size w was used to compute the texture measure at each pixel as in the other compared methods. In the PanTex algorithm we used a square root of contrast measure derived from the GLCM matrix. This contrast measure can actually be computed without explicit calculation of the GLCM matrix. The GLCM contrast measure was estimated within a window of arbitrary size w, such that displacement vectors of the length $\frac{w}{3}$ were determined by all pixels on a circle of radius $\frac{w}{3}$.

To compute the LBP local variance and the AMF measures we used a Matlab code available online[6]. In the LBP we used the square root of local variance computed as a variance of $4(w-1)$ equally spaced point samples on a circle of radius $\frac{w}{2}$. In the AMF approach we set the largest period of a sinusoid in the Gabor filters to $3w$ pixels. Note that similar to the MTC operator, all the algorithms, after the small modifications described above (except for AMF), are bias invariant, invariant to signal inversion and proportional to signal magnitude.

3.1 Qualitative Comparison

The contrast descriptors obtained using the compared transformations are shown in the first three columns in Fig. 3. All the descriptors have high values in textured areas and low values in smooth areas. However, contrary to the MTC operator, the other approaches yield also high responses at isolated features (edges, ridges, peaks) that do not belong to texture. The PanTex descriptor partially succeeded to suppress isolated curvilinear structures.

To better visualize the accuracy in texture localization, the texture descriptors were superimposed on the enlarged part of the satellite image in the forth column of Fig. 3, where the contrast of red tones is proportional to the values of the descriptors. Since the distribution of descriptor values is strongly bimodal, one can distinguish two major levels of texture descriptors, low and high, that appear as a gray-reddish and saturated red overlayed on the original image. As can be seen from these images, another advantage of the MTC operator is that it does not extend the borders of texture regions as other methods do. The CC method does not extend borders of texture regions, instead it generates a halo near texture borders and around individual features. This effect does not occur in the original version of the CC method, in which disjoint/overlapping block processing was performed (which, however, would not allow accurate texture localization). The LBP method also leaves rings around individual trees due to high response at distances equal to the radius of the circle used to compute the variance of samples on it.

[6] Matlab sources are available at `http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab` and `http://cvsp.cs.ntua.gr/software/texture/`

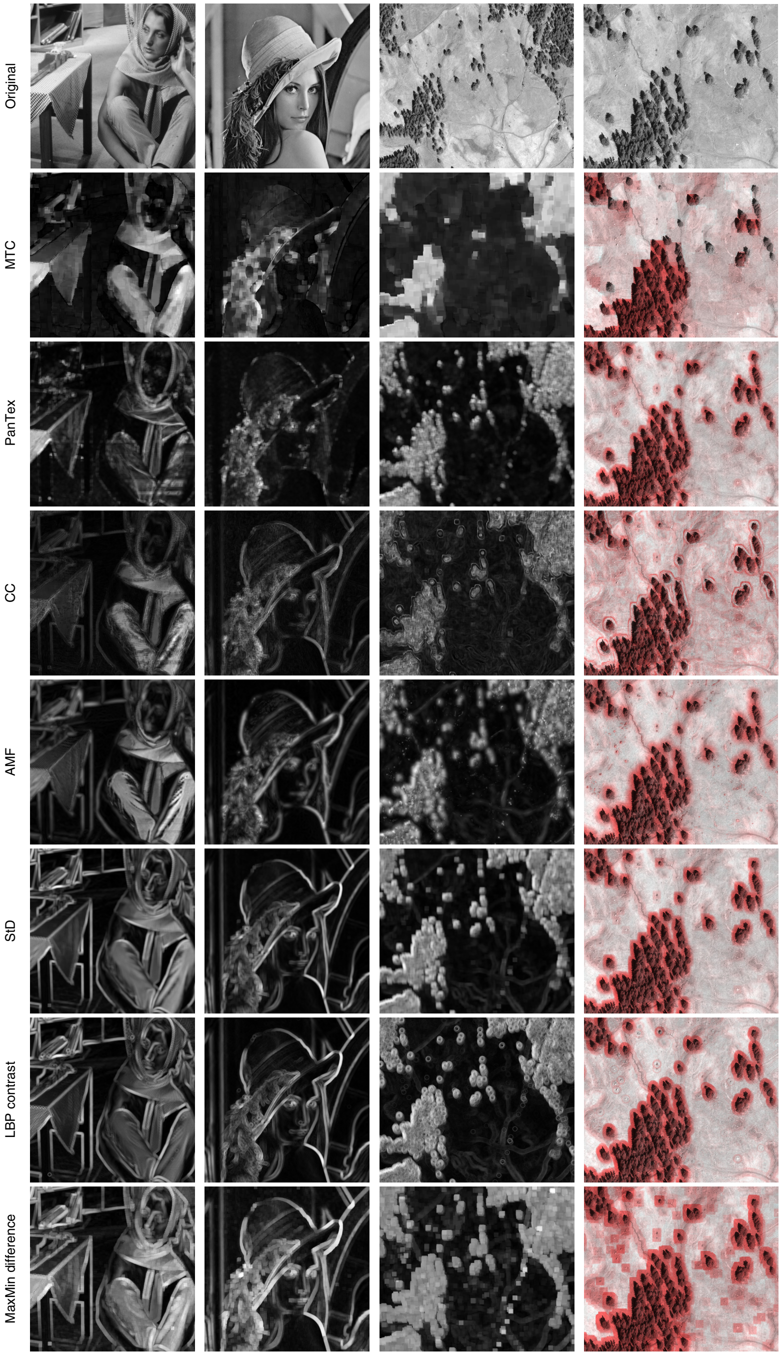


Fig. 3. Comparison of texture contrast descriptors. The first two original images are of size 512x512 pixels; the third image is the same as in Fig. 2(a); the forth image is an enlarged part of the third image. $w = 10$ for the first two images and $w = 30$ for the satellite images. Further details can be found in Sec. 3 and Sec. 3.1.

3.2 Quantitative Comparison

In order to quantitatively compare the ability of the methods to distinguish between texture and non-texture areas we quantify separability between distributions of texture descriptors in these areas. We used the Fisher criterion [15] that measures the distance between distribution means relative to their compactness. The criterion is given by $\frac{(\mu_1-\mu_2)^2}{\sigma_1^2+\sigma_2^2}$, where μ denotes a class mean and σ^2 denotes a class variance. Since ground truth data is required to define textured and non-texture regions, we created an artificial data set of gray-scale images along with corresponding masks of texture regions and non-texture areas, whereby the last also include individual features.

The data set consists of 100 images of 300x300 pixels with circular texture clusters and individual features (clusters may overlap, see the upper-left image in Fig. 5). The number of clusters and their diameters were chosen uniformly randomly and varied from 2 to 4 and from 60 to 120 pixels, respectively. The diameter of both individual features and texture details was 5 pixels. Texture details within clusters were placed at positions on a regular grid with random Gaussian offsets. The distance between nodes of the regular grid was set to 9 pixels. The amplitude of texture details and individual features varied randomly with normal distribution. White noise was added followed by smoothing with an averaging filter with a 3x3 kernel. The standard deviation of the noise was equal to one third of the amplitude of the texture details.

In the first two experiments we set the mean amplitude of individual features equal and triple, respectively, of the amplitude mean of texture details. Fig. 4(a, b) show the resulting separability measure for the size parameter w from 10 to 70 pixels. The figures show superiority of the MTC operator when discriminating non-texture regions from texture areas. These figures also reveal a high degree of immunity of the MTC approach to individual features with high magnitude.

In the third experiment we restrict the class of non-texture areas to areas adjacent to texture regions and to individual features including their neighborhood. Fig. 4(c) shows separability between such restricted non-texture areas and texture regions when mean amplitude of texture details and individual features were equal. A comparison of Fig. 4(c) and Fig. 4(a) confirms that the superiority in the performance of the MTC operator stems from its ability to distinguish texture from isolated features as well as from regions adjacent to texture borders.

4 Extraction of Isolated Features: The Morphological Feature Contrast (MFC) Operator

Using the ideas underlying the MTC operator, below we propose a Morphological Feature Contrast (MFC) operator that extracts isolated structures while suppressing texture details of textured background. Using alternating morphological filters, upper and lower texture envelopes were estimated in the MTC approach. To extract bright or dark individual features, we suggest using the difference between the original signal and one of its envelopes, as defined in the

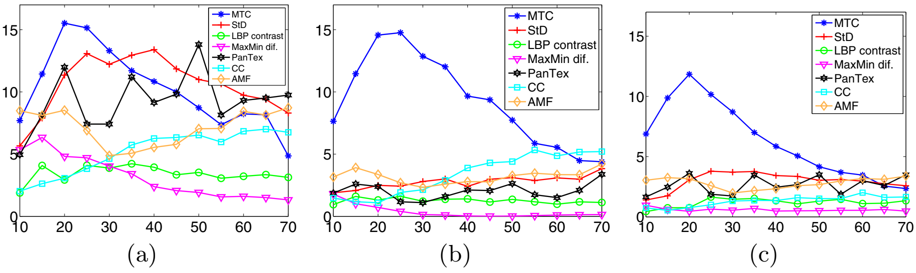


Fig. 4. (a) and (b): The measure of separability between texture and non-texture regions as a function of the scale parameter w. The mean amplitude of individual features is equal to the amplitude of texture details in (a) and tripled in (b). (c) The measure of separability of texture regions from areas around individual features and smooth areas adjacent to texture borders. The mean amplitude of individual features equals the amplitude of texture details.

following equations

$$\psi^{+}_{\mathrm{MFC}}(f) = |f - \gamma_{r_2}\varphi_{r_1}(f)|^{+} \ , \tag{2}$$

$$\psi^{-}_{\mathrm{MFC}}(f) = |\varphi_{r_2}\gamma_{r_1}(f) - f|^{+} \ . \tag{3}$$

If one is interested in both dark and bright features, $\psi_{\mathrm{MFC}} = \psi^{+}_{\mathrm{MFC}} + \psi^{-}_{\mathrm{MFC}}$ should be used. We call ψ^{+}_{MFC} and ψ^{-}_{MFC} white and black MFC, respectively. Similar to the MTC transformation, ψ_{MFC} is bias invariant, invariant to signal inversion, and proportional to signal magnitude. The MFC operators applied to a 1D artificial signal are illustrated in Fig. 1. The size r_1 of the SE should be greater than the maximal distance between details of texture to be suppressed. The size r_2 should be greater than the size of features to be extracted. If not stated otherwise, we use equal sizes for both SEs.

It can be shown that ψ_{MFC} is equal to $\max(\varphi\gamma(f), f) - \min(\gamma\varphi(f), f)$, while ψ^{+}_{MFC} and ψ^{-}_{MFC} are equal to operators that were already defined in [16] as $f - \min(\gamma\varphi(f), f)$ and $\max(\varphi\gamma(f), f) - f$, and were used for detection of defects in the noisy background of a metallic surface.

Fig. 5 and Fig. 2(d) show examples of the MFC operators ψ_{MFC} and ψ^{-}_{MFC} applied to gray-scale images. One can observe that various individual features/objects were highlighted while texture areas were simultaneously suppressed. For example, in the middle image in Fig. 5 the forest texture area and the texture of the village center on the right were suppressed, while isolated buildings outside the dense village center were preserved in the output image.

The MFC operator is capable of suppressing texture areas even if composed of details of higher magnitude and similar shape in relation with the magnitude and shape of individual features. Although several methods were developed to extract object boundaries (edge features) from textured background, we are not aware of other techniques that perform qualitatively similar to the MFC when extracting blob-like features (as well as features of an arbitrary shape).

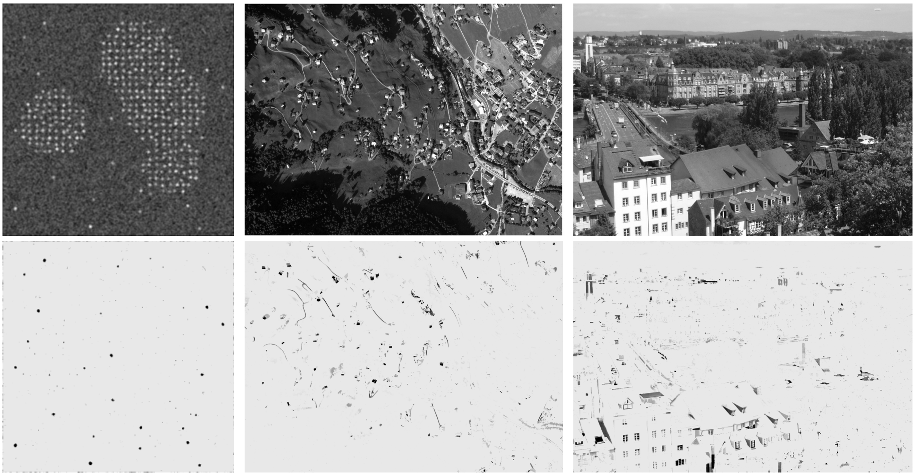


Fig. 5. First row: 312x312 artificial, 2570x1870 satellite[4], and 1010x690 digital camera images. Second row: individual features extracted by means of the MFC operator ψ_{MFC}[5]. $r_1 = 30, r_2 = 10$ for the artificial image; $r_1 = r_2 = 90$ for the satellite image, and $r_1 = r_2 = 25$ pixels for the digital camera image.

4.1 Application of the MFC Operator to Extraction of Isolated Linear Features

Above we have shown that the MFC operators are capable of extracting features of different types with width smaller than r_2. Specific features can be extracted by a sequence of standard morphological transformations, with the structuring element shaped similarly to features. Here we illustrate advantages of the use of the MFC within such a sequence for the case of linear features.

The remotely sensed images in Fig. 6 (left) contain rectangular structures composed of linear walls, which used to be livestock enclosures. A black top-hat transform removes background and emphasizes small dark structures in the image. A subsequent filter γ_{lin} obtained by the point-wise maximum of morphological openings with linear SE at different orientations extracts narrow linear structures longer than the length of the structuring element. In the resulting image shown in Fig. 6 (middle) linear walls were highlighted but also texture details were emphasized. Furthermore, an appropriate threshold setting is required to obtain a binary map of features.

To remove texture while keeping isolated features, the MFC operator ψ^+_{MFC} may be applied[7] prior to γ_{lin}. This sequence of operators completely removes most texture details. Non-zero pixels of the resulting image are shown in black in Fig. 6 (right). No threshold selection was required to obtain the binary map of linear features. We are currently evaluating the potential of using the

[7] The ψ^-_{MFC} could also be directly applied to the initial image.

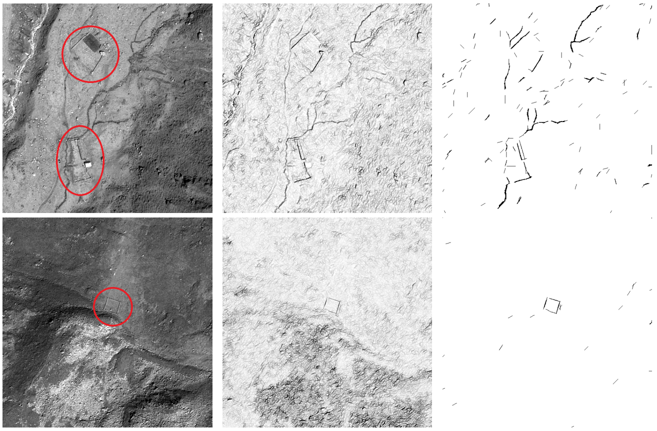


Fig. 6. Left: Satellite images[4] of 600x600 pixels size with man-made structures (linear walls). Middle: Black top-hat transform followed by an opening filter γ_{lin} obtained by the point-wise maximum of morphological openings with linear structuring elements at different orientations[5]. Right: Black top-hat transform followed by ψ^{+}_{MFC} and subsequent γ_{lin}. Non-zero pixels are shown in black. Parameters: 5x5 SE in the top-hat operator, $r_1 = 5, r_2 = 10$ in ψ^{+}_{MFC}; the length of the linear SE in γ_{lin} equals 15 pixels.

MFC operators for extraction of linear features within the Silvretta Historica project[8].

4.2 Extension of the MFC Operator to Vector-Valued Images

In this section we generalize the MFC operators from gray-scale to vector-valued discrete images, where each pixel is attributed by a vector of values. A multispectral image is an example of such a vector-valued image. The problem with extending morphological operators to vector-valued images lies in the lack of a natural ordering of vectors. However, some morphological transformations defined in terms of arithmetic differences between morphological operators can naturally be extended to vector-valued images without the need to chose a vectorial order. Examples of such extended transformations were recently proposed for morphological gradient and for top-hat in [18], [19]. Using similar ideas we derive an extended version of the MFC operators below.

The MFC operator defined in Eq. (3) can be rewritten in the following form $\psi^{-}_{\text{MFC}}(f) = |\varepsilon_{r_2}\delta_{r_2}\delta_{r_1}\varepsilon_{r_1}(f) - f|^{+}$ where δ and ε denote morphological dilation and erosion, respectively. r_1, r_2 denote sizes of SEs $B_p^{(1)}$ and $B_p^{(2)}$, respectively, centered at p. Omitting the details of derivation, the last equation can further be transformed to

$$[\psi^{-}_{\text{MFC}}(f)](x) = \min_{k \in B_x^{(2)}} \max_{j \in B_k^{(3)}} \min_{i \in B_j^{(1)}} |f(i) - f(x)|^{+} \ , \tag{4}$$

[8] Details on this project can be found in [17].

where $B^{(3)}$ denotes the structuring element $B^{(1)}$ dilated by $B^{(2)}$. Similarly,

$$[\psi^{+}_{\mathrm{MFC}}(f)](x) = \min_{k \in B^{(2)}_x} \max_{j \in B^{(3)}_k} \min_{i \in B^{(1)}_j} |f(x) - f(i)|^{+} \ . \tag{5}$$

We now define a new vectorial MFC operator $\psi_{\mathrm{MFC}}(\bar{f})$ that applies to vector-valued images $\bar{f}$. We replace the non-negative difference between intensity values in the last two equations by a suitable metric distance D between vectors,

$$[\psi_{\mathrm{MFC}}(\bar{f})](x) = \min_{k \in B^{(2)}_x} \max_{j \in B^{(3)}_k} \min_{i \in B^{(1)}_j} D(\bar{f}(x), \bar{f}(i)) \ . \tag{6}$$

In contrast to $\psi^{+}_{\mathrm{MFC}}(f)$ and $\psi^{-}_{\mathrm{MFC}}(f)$, the $\psi_{\mathrm{MFC}}(\bar{f})$ operator extracts both dark and bright structures when applied to multispectral images. If one is interested in extraction of either dark or bright features only, pseudo-distances may be used. For example, instead of using D_{∞} distance, pseudo distances defined by $D^{+}_{\infty}(\bar{f}, \bar{g}) = \max_i |f_i - g_i|^{+}$ and $D^{-}_{\infty}(\bar{f}, \bar{g}) = \max_i |g_i - f_i|^{+}$ may be employed.

Vectorial operators may be preferable to independent processing of channels of a vector-valued image followed by integration of the results. On the other hand, in many cases, independent processing produces similar or even better results due to smaller levels of output noise. A comparative evaluation of the vectorial MFCs is beyond the scope of this paper.

5 Summary

We have shown how alternating morphological filters can be used to design operators for detection of texture regions and isolated features. The comparison of the morphological texture contrast (MTC) operator based on visual inspection and quantitative experiments, reveals its superiority over other methods used for texture detection. This operator is very attractive for various applications due to its ability to discriminate texture from isolated features irrespectively of their high magnitude, good localization properties, and simplicity. The complementary morphological feature contrast (MFC) operator was proposed for extraction of isolated features in images containing also texture background. We show that the MTC and the MFC have a similar structure and are of special interest where it is important to distinguish isolated features from texture details. An extension of the MFC operator was derived that allows the MFC operator to work directly with vector-valued images. We have also proposed a simple scheme based on the MFC operator for detection of isolated linear structures.

Acknowledgments. This work was funded by the Interreg IV Program "Alpenrhein - Bodensee - Hochrhein" and by the Zukunftskolleg, University of Konstanz. It was also supported by the DFG Research Training Group GK-1042 "Explorative Analysis and Visualization of Large Information Spaces."

References

1. Dinstein, I., Fong, A., Ni, L., Wong, K.: Fast discrimination between homogeneous and textured regions. In: Proc. of Int. Conf. on Pattern Recognition, Montreal, Canada (1984)
2. Gonzalez, R.C., Woods, R.E.: Digital Image Processing (2001)
3. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24, 971–987 (2002)
4. Kokkinos, I., Evangelopoulos, G., Maragos, P.: Texture analysis and segmentation using modulation features, generative models and weighted curve evolution. IEEE Trans. Pattern Analyis and Machine Intelligence 31, 142–157 (2009)
5. Pesaresi, M., Gerhardinger, A., Kayitakire, F.: A robust built-up area presence index by anisitropic rotation-invariant textural measure. IEEE Journal of Selected Topics in Applied Earth Observation an Remote Sensing 1 (September 2008)
6. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Trans. Systems Man and Cybernetics 6, 610–621 (1973)
7. Bergman, R., Nachlieli, H., Ruckenstein, G.: Detection of textured areas in natural images using an indicator based on component counts. J. Electronic Imaging 17(4) (2008)
8. Karu, K., Jain, A.K., Bolle, R.M.: Is there any texture in the image? Pattern Recognition 29, 1437–1446 (1996)
9. Zingman, I., Saupe, D., Lambers, K.: Morphological operators for segmentation of high contrast textured regions in remotely sensed imagery. In: Proc. of the IEEE Int. Geoscience and Remote Sensing Symposium, Munich, Germany (July 2012)
10. Serra, J.: Image Analysis and Mathematical Morphology, vol. 2: Theoretical Advances. Academic Press (1988)
11. Grigorescu, C., Petkov, N., Westenberg, M.: Contour detection based on nonclassical receptive field inhibition. IEEE Trans. Image Processing 12, 729–739 (2003)
12. Dubuc, B., Zucker, S.: Complexity, confusion, and perceptual grouping. Part II: Mapping complexity. J. Mathematical Imaging and Vision 15, 83–116 (2001)
13. Serra, J., Vincent, L.: An overview of morphological filtering. Circuits, Systems, and Signal Processing 11, 47–108 (1992)
14. Soille, P.: Beyond self-duality in morphological image analysis. Image and Vision Computing 23, 249–257 (2005)
15. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, Inc. (1990)
16. Salembier, P.: Comparison of some morphological segmentation algorithms based on contrast enhancement - application to automatic defect detection. In: European Signal Processing Conference, Barcelona, Spain, pp. 833–836 (September 1990)
17. Lambers, K., Zingman, I.: Towards detection of archaeological objects in high-resolution remotely sensed images: the Silvretta case study. In: Proc. of Computer Appl. and Quantitative Methods in Archaeology, UK (2012) (in press)
18. Evans, A.N., Liu, X.U.: A morphological gradient approach to color edge detection. IEEE Trans. Image Processing 15, 1454–1463 (2006)
19. Hanbury, A.: The morphological top-hat operator generalised to multi-channel images. In: Proc. of the Int. Conf. on Pattern Recognition, pp. 672–675 (August 2004)

Estimation of Separating Planes between Touching 3D Objects Using Power Watershed

Clara Jaquet[1], Edward Andò[2], Gioacchino Viggiani[2], and Hugues Talbot[3]

[1] Université Paris Est Créteil, Institut Supérieur de BioSiences, Créteil, France
[2] Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge,
Equipe A3SI, ESIEE, Noisy-le-Grand, France
[3] Grenoble-INP / UJF-Grenoble 1 / CNRS UMR 5521, Laboratoire 3SR, Grenoble, France

Abstract. The problem of separating touching or overlapping objects is classical in imaging. Many solutions have been proposed in 2D. While similar, the problem in 3D has differentiating features: apparent overlap due to projection effects does not exist, but real or apparent interpenetration can occur only due to either physical particle fusion or partial volume effects. Often the ability to separate objects logically is sufficient, however sometimes finding the orientation of tangent separating plane is useful. In this article, we propose a method based on power watershed for separating 3D touching objects and estimate a precise separating plane. Power watershed is used in two steps, first to obtain individual object identification, and in a second step to allow sub-voxel accuracy in the plane fitting procedure. We show that our approach is much more precise than a simple segmentation. We illustrate this in an application involving the shearing of a sample of sand grains imaged in various configurations by micro-CT tomography. Our technique measures the orientation of the contacts between grains, a quantity that is explicitly used in soil mechanics modeling, but which has up until now been difficult to measure from experiments.

Keywords: Segmentation, random walker, orientations, micro-tomography.

1 Introduction and Motivation

In this article, we aim to accurately measure the contact orientations between touching 3D particles.

Our motivation is to enable some experimental measurements in soil mechanics. When granular materials such as sand are mechanically loaded, strain localisation can manifest itself as shear banding which leads to rapid failure of the material, associated with catastrophic events such as landslides. In order to better understand the process of strain localisation in a granular material such as sand, it must be studied at the scale at which strain localises; for sands this is likely to be the grain scale. Historically, micro-mechanical information has been difficult to measure in real experiments so this information has generally been obtained by numerical simulations on large numbers of particles. Recent advances in X-ray tomography allow individual sand grains to be imaged in 3D, paving the way for grain-scale measurements on real granular materials. A full micro-mechanical description of the kinematics occurring at the grain scale needs

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 452–463, 2013.

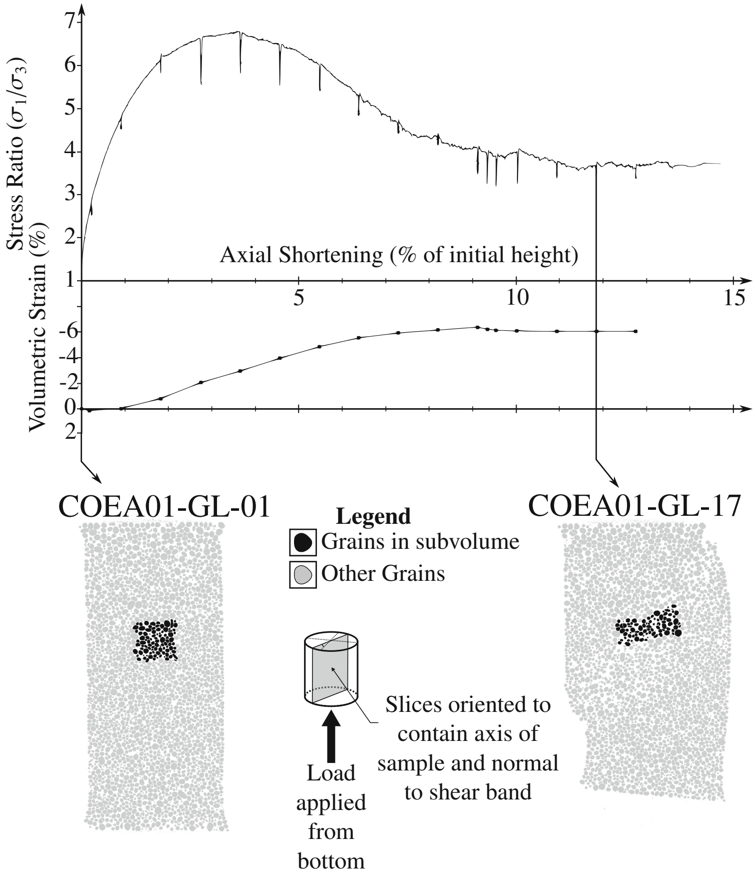


Fig. 1. Top: Stress ratio and volumetric strain plotted against the axial shortening imposed on the sample tested. Stress relaxations are visible when loading has been stopped to allow an x-ray tomography to be performed. Bottom: Vertical cross sections from sample COEA01, oriented so as the contain the axis of the sample as well as the normal to the shear band which can be seen on the right. These slices have been taken along the *x* direction, (*z* is the vertical axis)

to include *grain kinematics* (rotations and displacements of grains), as well as *contact kinematics* (the gain or loss of contacts, and the orientation of those contacts).

Fig 1 illustrates the process of strain localisation: vertical sections from 3D images of a sample show that a small subvolume of grains is deformed by a shear band that develops in the sample. In such a sample there are more than fifty thousand grains, so automatically and accurately segmenting each grain as well as providing unbiased estimates of grain contact orientation is essential for a complete micro-mechanical description of the processes at work.

In 2D images, separating binary overlapping convex objects is a common and classical problem in image analysis. It has been presented to students of Mathematical Morphology (MM) for a long time [3], and is typically solved with distance transforms [18] and watershed [4]. It is one of the most well-known application of MM in general [10,11]. This simple distance-based solution has many drawbacks. Proposed improvements have led to many fruitful concepts in MM, such as h-extrema [23], elliptical distance transforms [21] and conditional bisectors [14,22,7].

In 3D the problem is different. In 2D the need to separate overlapping objects is common due to projection effect, whereas in 3D this effect does not exist. However there exist some physical cases where interpenetrating (quasi) convex objects are grown or fused together such as sintering processes in materials science. Furthermore, when objects are small, lack of resolution or diffusion in some imaging modalities may cause an effect known as *partial volume effect* to become prevalent [5]. This may cause 3D particles to appear fused rather than simply touching.

Separating fused objects in 3D is conceptually no different than in 2D, and indeed most methods developed for 2D readily carry over to higher dimensions. However the problem we study in this article is somewhat different again. In many applications, segmenting the fused objects is sufficient, allowing separated objects to be counted and measured. Nonetheless, when two objects are touching physically, it is sometimes useful to study the contact itself, in particular the spatial orientation of the plane of contact, when such a concept can be defined. We consider a correctly oriented separation plane between two grains to be a good approximation of a mechanically relevant contact orientation. A reliable and accurate way to automatically measure orientation of the contact surface between objects is therefore required.

The rest of the paper is organized as follows: in section 2 we provide the necessary background notions. In section 3 we describe our proposed segmentation and separating plane orientation estimation procedure. In section 4 we present results on simulated touching spherical particles. In section 5 we present results on real data, first on physical spherical beads, and then on actual sand grains. We conclude by showing that our procedure is less biased than previous methods for separating grains.

2 Background Notions

Here we present the necessary notions for our proposed method.

2.1 Watersheds

Watershed is the de-facto standard procedure for segmentation in MM [16]. Watershed as a segmentation procedure was proposed at the end of the 1970s [4], however fast procedures enabling its efficient computation were only proposed in the 1990s [24,15]. Various types of watershed algorithms have been proposed. We mention some that produce labeled regions with a thin inter-pixel boundaries like Meyer's algorithm [15], some that produce a thick pixel boundary like the topological watershed [8], and some that make explicit use of the graph structure of the underlying image like watershed cuts [9]. In the latter case, the resulting separating cut or boundary is located inter-pixel.

2.2 Power Watershed

Recently a new type of watershed was introduced, the power watershed [6]. It is defined as the epi-convergence of the result of the optimization of the following energy:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \sum_{e_{ij} \in E} w_{ij}^{p} |x_i - x_j|^q + \sum_{v_i \in V} w_i^p |x_i - y_i|^q, \tag{1}$$

where (V, E) is a graph consisting of an edge set E linked by vertices belonging to V. Vertex i is denoted v_i ; the edge linking v_i to v_j is denoted e_{ij}. The w_{ij} and w_i are respectively pairwise and unary edge weights that are given. The y_i are unary references that are given (e.g. seeds), and the x_i are vertex value belonging to $\mathbf{x}$, the set of all the vertex values that we seek to optimize. In imaging, the pixel locations correspond to vertices and the edges represent the underlying neighborhood connectivity. In this paper we consider the nearest neighbour, i.e. the 6-connectivity for 3D. In a Maximum A Posteriori (MAP) interpretation, similarly to Graph Cuts, the first sum over edges is interpreted as a regularization or smoothness term, while the second sum over vertices is a data fidelity term. In particular, this data term can be easily used to enforce seeds for image segmentation, i.e. regions where the labelling $\mathbf{x}$ is known and fixed.

Finally, in (1), p and q are integers with specific interpretations. In particular, when q is set to 2, for every finite p, the solution to (1) represents the energy of a Random Walker (RW) [12]. As p tends to infinity, the resulting $\mathbf{x}_\infty$ epi-converges to a unique solution, which is termed the power-watershed. Indeed, this $\mathbf{x}_\infty$ has many of the properties of a usual watershed. In particular if the data fidelity represents binary seeds, then $\mathbf{x}_\infty$ is a binary labelling everywhere equal to the result of a watershed cut, except on separating plateaus. These are the regions of constant edge weighting that are contested between the two labels. On these edge plateaus, the solution is that of a random walker.

In [6], an efficient algorithm is provided to compute this solution without resorting to a convergence process. On contested plateaus, as in the RW result, the values of $\mathbf{x}_\infty$ can be interpreted as the probability of a discrete random walker moving from vertex to vertex along the edges of the graph to reach one label vs. the others. In the case with arbitrary n markers, one needs to compute $n - 1$ probability fields, as they sum to 1. Specifically, in the case of two markers A and B, if the probability of reaching A from some point is p, then the probability of reaching B is $1 - p$, so computing a single probability field is indeed sufficient.

2.3 Objects Separation with Classical and Power Watersheds

The procedure for object separation with classical watershed is well known, but we describe it here for reference. We assume 3D volume images and a pair of interpenetrating, approximately convex objects.

1. Given a binary image I containing the objects, a distance transform $E(I)$ is computed inside all objects. In this article we used the Euclidean distance transform [19] because it is isotropic.
2. The maxima of this distance transform $\mathrm{MAX}(E(I))$ are used as seeds. To account for boundary noise, more robust h-maxima can be used instead [20].

3. The Euclidean distance map image is inverted, and a watershed transform is computed on it $\mathrm{WS}(-E(I))$, stemming from the seeds computed at step 2.

This procedure finds a satisfactory separating surface, even if the objects are of different size, if the objects interpenetration is not too deep, which will be the case in our application. Depending on the watershed transform used, the output may be a labeled image, in which case the watershed surface is assumed to be located between neighboring voxels with different labels, or an actual discrete surface composed of voxels.

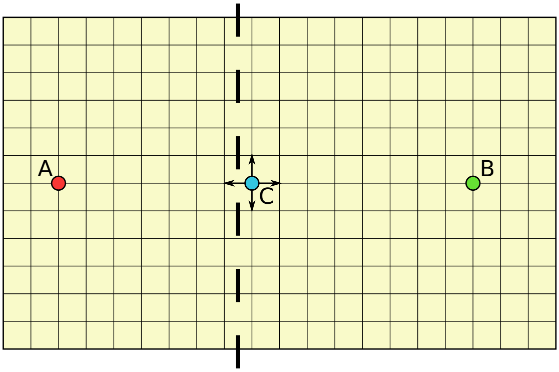


Fig. 2. Object separation with power watershed/random walker. Assuming a constant edge plateau, except on the thick vertical dotted line, where edges that are crossed are given a high weight, i.e. are difficult to cross. A and B represent labels. We estimate the probability of a random walker in C to move to the west as opposed to all the other directions (north, south or east). We see that locally the probabilities are identical because all neighboring edges at this point have the same weight. However globally, there are many ways for a random walker to come from C to B. whereas there is only one way to go from C to A. Therefore, random walker probabilities must be high east of C, and then drop rapidly through the vertical dotted line.

In the case of the power watershed, the procedure is different. Indeed, seeds are still needed, and can be extracted from the maxima of the Euclidean distance map, however the inverted distance map is not necessary for the computation of the labelling $\mathbf{x}$ described in section 2.2. Indeed, if we consider the binary object as a contested plateau, then the power watershed reduces to the computation of a pure random walker stemming from the disjoint seeds. In this case the computation of $\mathbf{x}$ on the plateau exhibits a sharp variation when crossing the boundary from one object domain to the next. This is due to the fact that a random walker is much more likely to reach the nearest seed in its domain of origin rather than cross the domain to the other seed through the small aperture between them. Therefore, to first order approximation, the resulting labelling $\mathbf{x}$ is almost binary on most of the contested plateau, with values very close to 1 near one seed, and close to 0 near the other. However a random walker placed close to the boundary between the two interpenetrating object is approximately as likely to reach either seed, and so the values of $\mathbf{x}$ in this area are close to 0.5. To find a separating surface between the two objects, we can therefore consider the 0.5-isosurface on $\mathbf{x}$. It has the advantage of a more precise location, benefiting from sub-pixel accuracy.

3 Methodology

This section details the method by which the surface that separates two touching objects in 3D can be accurately oriented. We describe our technique assuming a pair of touching objects. Due to unavoidable partial volume effects, these two objects appear slightly fused.

In this paper we present data from two sources: from simulations of spheres (section 4), and from 3D images of experiments of a granular material. The simulation directly produces a pair of contacting grains, whereas the images from experiments contain several thousands of grains; experimental data are first segmented and labelled with the standard distance-based approach to extract pairs of grains.

Once a 3D image with two grains is obtained, one seed is identified inside each grain. In the simulations we simply use the centre of the spheres. In the real data, seeds inside each grain are computed from the 3D Euclidean distance map of the fused particles and h-maxima. The power watershed is then computed using these two seeds. As explained above, the result of this is a map for each seed describing the probability of each voxel of belonging to one or the other of these seeds. Points on the 0.5 isosurface are then identified, by interpolating the probability along the edge linking two neighboring voxels with differing labels. We record the list of these points and their coordinates and we fit a plane through these points using Principal Component Analysis (PCA). The normal to this plane is recorded as the orientation of the contact surface (or tangent) between the two grains.

For comparison, we followed a similar procedure with several standard watersheds, namely the Meyer algorithm, the topological watershed and watershed cuts. We also tested a proprietary algorithm. In cases were the algorithm yields a labeling with an inter-pixel separating surface, we took the midway point between pairs of pixels with different labels as being part of the separating surface. In cases where the algorithm yields a voxel-thick surface, we recorded the center of the voxels belonging to the surface. In all cases we performed a PCA analysis on the resulting point cloud to fit a plane and identified its normal as the orientation of the surface of contact between the two touching objects.

4 Testing on Simulations

In order to test our contact orientation measurement procedure, simulations of 3D Euclidean balls of radius 10 voxels were placed in a random orientations so that their centres are 20 voxels apart (so that the spheres are just touching). These balls were digitized as binary objects.

Balls are used in this case since they have the unique property that the vector connecting their centres (“branch vector”) has the same orientation as the normal to the contact surface between the balls, providing ground truth for comparing measured orientations of the separation plane. Using the centres of the balls as seeds, the balls are then segmented using the technique outlined in the previous section. Fig. 3 shows the result of the segmentation and the computed location of some points on the 0.5-isosurface defining the contact plane.

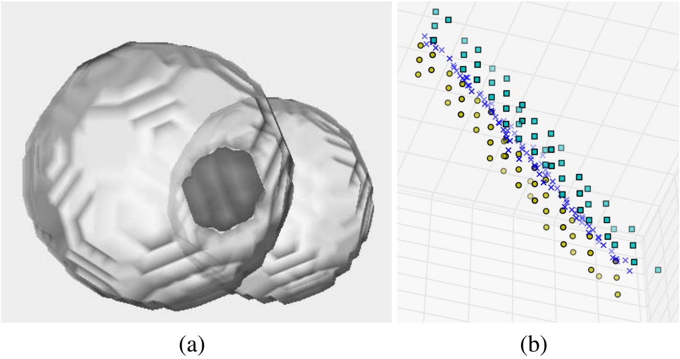
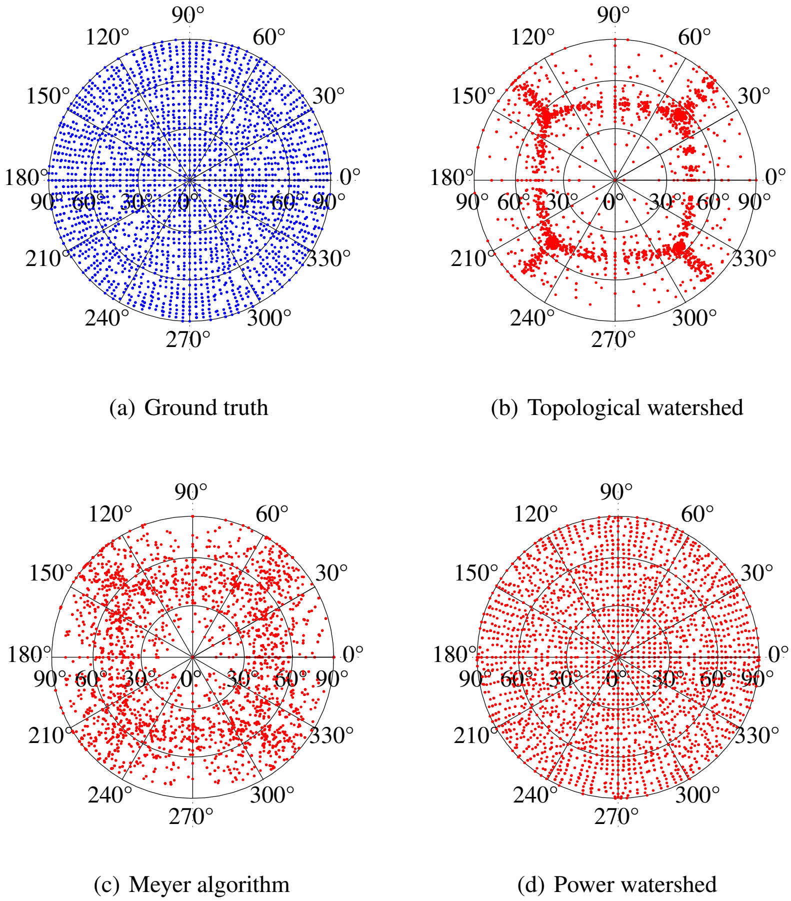


Fig. 3. Two touching spheres. (a) shows the area of overlap in dark grey and is close to a discrete disk. (b) shows the voxels on either side of the 0.5-isosurface in square or disk shape. In cross shape the location of the interpixel 0.5-isosurface. All points are very close to being located on the same plane.

The computed orientation of the contact plane as an (x, y, z) unit vector between the two balls can then be projected onto the unit z-positive half-sphere, since we cannot distinguish between either normals to the plane. To represent this projection, two numbers are sufficient, equivalent to a longitude and latitude. To represent these projections on 2D paper, we used the Lambert polar equal-area projection [13], which is easy to compute. If orientations are identically and independently distributed on the unit half-sphere, this projection yields a point distribution on a disk indistinguishable from a 2D Poisson point process [17]. However, in the case of this simulation, because we only simulated balls with centers located on a voxel, there is a finite number of possible orientations, and so our simulation shows a discrete bias.

Indeed, Fig 4(a) shows the ground truth of evenly-distributed orientations obtained with a sufficient number of randomly places spheres. Figure 4(b) shows the orientations of the separation planes obtained with a standard topological watershed. Although the distribution of the orientations should be the same, it is clear that the topological watershed introduces strong artefacts into the measured orientations, aligned at 45° to the coordinate system. Figure 4 (c) shows the orientations of the separation planes obtained with the Meyer algorithm, in which the definition of the delimition between the two particles is *interpixel*, showing some improvement, however insufficient for our purpose.

Figure 4(d) shows the distribution of orientations obtained with the measurement of the orientation of the separation place with the random-walker approach described above; it is obvious that the distribution of contact orientations obtained with this method is much closer to the true value. Indeed, the median orientation absolute error is 0.61°, which is very small. We now try our method on real data.



(a) Ground truth

(b) Topological watershed

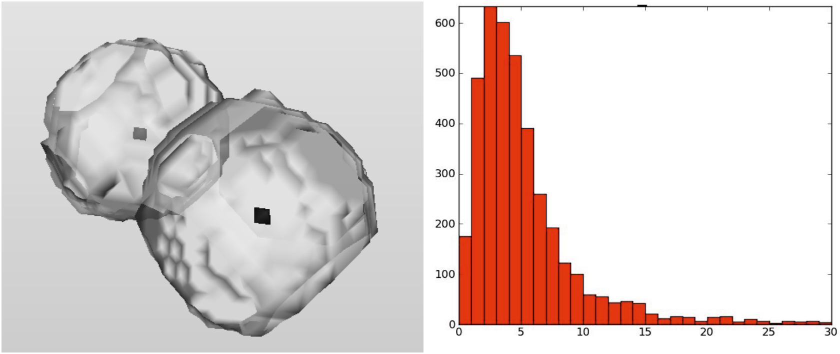
(c) Meyer algorithm

(d) Power watershed

Fig. 4. Equal area projection of orientations of separation planes from simulated spheres

5 Testing on Real Data

The experimental data in this work come from tomographic images acquired on mechanical experiments on small samples of sand. Samples are prepared by “pluviating” the granular material into a cylindrical, 22 mm height by 11 mm diameter latex membrane. Sample are installed into a specifically-designed pressure cell; the cell is then filled with water and pressurised. The cell allows the sample to be rotated while it is imaged by x-rays – meaning that a 3D image of the sample can be obtained by reconstruction. The sample can then be deformed axially, by advancing a piston under displacement control (and measuring the reaction force), and re-imaged in-situ.



(a) (b)

Fig. 5. Glass beads (the flat areas are due to contact with other beads). (a) Center of mass of each grain in dark grey. (b) Histogram of absolute angular deviation between branch and experimental vectors.

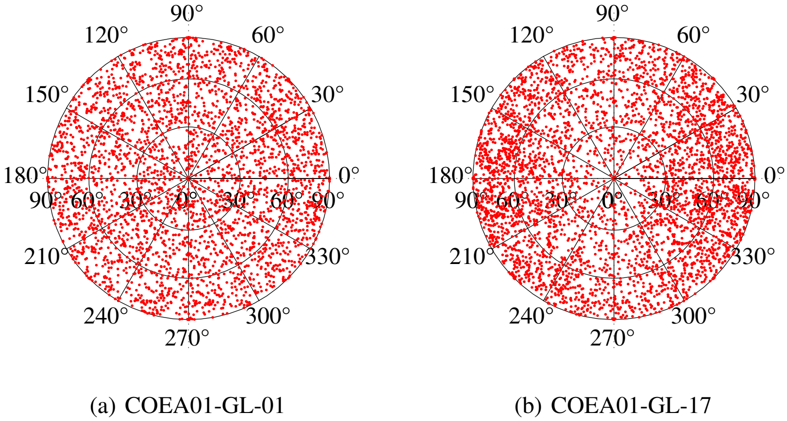
5.1 Spherical Grains

A first test case was performed on images coming from a sample (labelled ABEA02) of glass beads, which are close to perfect balls. This allows the same check to be performed as was done with the simulated balls: the branch vector can be calculated, providing a good estimation of the expected contact orientation. We computed the angle between branch and experimental vectors. On 4353 detected contacts we obtain a median error angle of only 4,27°, as show on Fig 5. Overall this approach allows a new level of precision to be obtained. This means that contact orientations can be measured even without access to the branch vector. We now apply our technique to a real sample of sand under stress.

5.2 Measuring the Change of Contact Orientation Due to Stress

Previous work has shown that individual grain kinematics (displacements and rotations) can be obtained by tracking them with a methods developed in [2]. Figure 1 shows the macroscopic stress reponse of the sample, and at the bottom shows two vertical sections through the sample COEA01-GL (a sample of Caicos ooids, a very rounded natural sand) in two different configurations. On the left the sample is under isotropic compression, before axial deformation. On the right, the sample has undergone considerable deformation, has reached and passed the peak in axial strength and has developed a shear band. In these images, the sub-selection of grains under study are shown in black, and the rest of the grains in the sample are shown in grey. Particle tracking was used to follow these grains from COEA01-GL-01 to COEA01-GL-17.

Figure 6 shows the orientations that can be obtained with the technique presented in this paper, in the two configurations shown in figure 1. Immediately observable is the much more uniform distribution of contact orientations, when compared to standard techniques. Furthermore, a change in the distribution of orientations is visible between the two granular configurations.



(a) COEA01-GL-01

(b) COEA01-GL-17

Fig. 6. Orientations of contacts between grains in configurations COEA01-GL-01 and COEA01-GL-17, as calculated using the power watershed

In order to make this distribution easier to study, the projected orientations are divided into equal-area bins on the plane onto which they have been projected. The number of projected points that falls inside each bin is recorded, and bin totals are normalised by the *median* bin value. In figure 7 the colourmap is centred on 1 such that segments in white are those in which the median number of points has been counted, red ones show ones where more than the median has been counted, and blue shows where fewer than the median have been counted. This figure reveals that there is a small amount preferential alignment of the orientations towards the cardinal directions (up, down, left, right, front, back), this is likely due to very small contact areas where the orientation can only be poorly defined.

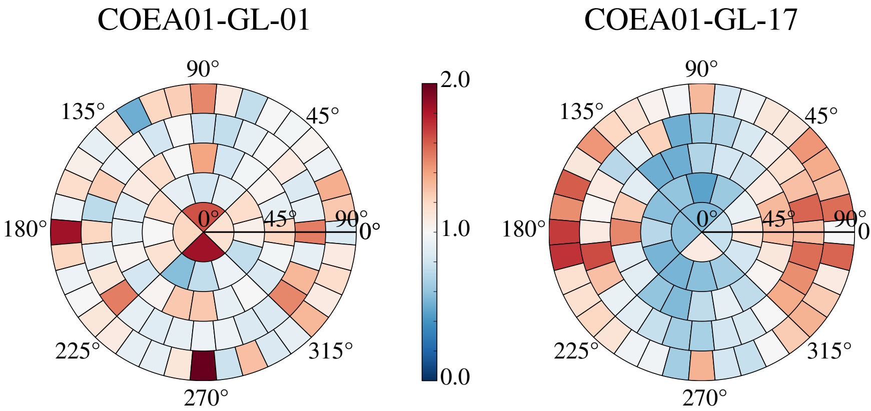


Fig. 7. Binned version of figure 6, where colours represent the number of projected points per bin divide by the mean number of projected points in all bins. Please refer to the online colour version for better interpretation.

From a mechanical point of view, a clear evolution in the distribution of contacts is visible: many contacts are gained in the x-direction and many are lost in the z-direction between configurations COEA01-GL-01 and COEA01-GL-17. Recalling that the sub-volume of grains on which this is calculated becomes part of the shear band that forms (whose normal is approximately [-1,-1,0]), this seems to indicate that grain-to-grain contacts are lost in the direction of principal stress applied to the sample (along the axis of compression). Conversely, contacts are gained in the direction in which the shear band is advancing.

.

6 Discussion and Conclusion

In this article, we have proposed a new method based on power watershed (PW) to compute the orientation vector of the contact plane between touching objects in 3D micro-CT images. We rely on a number of assumptions, namely that the objects are nearly convex, that as a result the contact is unique and somewhat regularized by partial volume effects. We consider touching objects in pairs using a method based on the standard watershed part of PW, and we compute the separating plane assuming the contact area is relatively small, using the random-walker part of PW. The contact plane is estimated with sub-voxel precision by estimating the equal-probability isosurface between the two objects.

We have compared this approach using various standard watersheds and a Euclidean distance map. On simulated data, as well as on real x-ray images of glass beads and sand grains, we have shown that our results are nearly free of artefacts, and show a very good isotropic spread of measurements. On simulated data, the measurement error is very low (0.61°), and on real glass bead data, for which we know the approximate contact plane orientation, the still small estimated median error is 4.27°. Consequently, we have been able to use this tool in a sand compression test, that shows conclusively that as pressure is applied, contact orientation anisotropy begins to develop in the sample.

This new technique paves the way towards a much more complete understanding of the micro-mechanics of real granular materials, enabling experimental data acquired on real granular media to be used in micro-mechanical models.

References

1. Andò, E., Hall, S.A., Viggiani, G., Desrues, J., Bésuelle, P.: Experimental micromechanics: grain-scale observation of sand deformation. Géotechnique Letters 2(3), 107–112 (2012b)
2. Andò, E., Hall, S.A., Viggiani, G., Desrues, J., Bésuelle, P.: Grain-scale experimental investigation of localised deformation in sand: a discrete particle tracking approach. Acta Geotechnica, 1–13 (2012)
3. Beucher, S., Gratin, C.: Micromorph reference manual, applications and solutions. Ecole des Mines de Paris (1989)
4. Beucher, S., Lantuéjoul, C.: Use of watersheds in contour detection. In: Int. Workshop on Image Processing, Rennes, France. CCETT/IRISA (September 1979)
5. Bloch, I.: Some aspects of dempster-shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account. Pattern Recognition Letters 17(8), 905–919 (1996)

6. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watersheds: A unifying graph-based optimization framework. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(7), 1384–1399 (2011)
7. Couprie, M., Coeurjolly, D., Zrour, R.: Discrete bisector function and euclidean skeleton in 2d and 3d. Image Vision Computing 25(10), 1543–1556 (2007)
8. Couprie, M., Najman, L., Bertrand, G.: Quasi-linear algorithms for the topological watershed. Journal of Mathematical Imaging and Vision 22(2-3), 231–249 (2005); Special issue on Mathematical Morphology
9. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: thinnings, shortest-path forests and topological watersheds. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(5), 925–939 (2010)
10. Russ, J.C.: The image processing handbook, 3rd edn. CRC Press (1999)
11. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice-Hall (2007)
12. Grady, L.: Random walks for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11), 1768–1783 (2006)
13. Hinks, A.R.: Map Projections, 2nd rev. edn. Cambridge University Press, Cambridge (1921)
14. Meyer, F.: Automatic screening of cytological specimens. Computer Vision, Graphics, and Image Processing 35(3), 356–369 (1986)
15. Meyer, F.: Un algorithme optimal de ligne de partage des eaux. In: Proc. 8ème Congrès Reconnaissance des Formes et Intelligence Artificielle, pp. 847–857. AFCET (November 1991)
16. Meyer, F., Beucher, S.: Morphological segmentation. Journal on Visual Communication and Image Representation 1(1), 21–46 (1990)
17. Miles, R.E.: On the homogeneous planar poisson point process. Mathematical Biosciences 6, 85–127 (1970)
18. Rosenfeld, A., Pfaltz, J.L.: Distance functions on digital pictures. Pattern Recognition 1, 33–61 (1968)
19. Saito, T., Toriwaki, J.I.: New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications. Pattern recognition 27(11), 1551–1565 (1994)
20. Soille, P.: Morphological Image Analysis, principles and applications, 2nd edn. Springer (2003) ISBN 3-540-42988-3
21. Talbot, H., Appleton, B.: Elliptical distance transforms and the object splitting problem. In: Talbot, H., Beare, R. (eds.) Proceedings of the 6th International Symposium on Mathematical Morphology, pp. 229–240. CSIRO Publishing, Sydney (2002)
22. Talbot, H., Vincent, L.: Euclidean skeleton and conditional bisectors. In: Visual Communications and Image Processing 1992, vol. 1818, pp. 862–873. SPIE, Boston (1992)
23. Vincent, L.: Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. IEEE Transactions on Image Processing 2(2), 176–201 (1993)
24. 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)

Efficient 1D and 2D Barcode Detection Using Mathematical Morphology

Melinda Katona and László G. Nyúl

Department of Image Processing and Computer Graphics
University of Szeged
Árpád tér 2, H-6720 Szeged, Hungary
katona.melinda@stud.u-szeged.hu, nyul@inf.u-szeged.hu

Abstract. Barcode technology is essential in automatic identification, and is used in a wide range of real-time applications. Different code types and applications impose special problems, so there is a continuous need for solutions with improved performance. Several methods exist for code localization, that are well characterized by accuracy and speed. Particularly, high-speed processing places need reliable automatic barcode localization, e.g. conveyor belts and automated production, where missed detections cause loss of profit. Our goal is to detect automatically, rapidly and accurately the barcode location with the help of extracted image features. We propose a new algorithm variant, that outperforms in both accuracy and efficiency other detectors found in the literature using similar ideas, and also improves on the detection performance in detecting 2D codes compared to our previous algorithm.

Keywords: barcode detection, morphological operations, bottom-hat filter, distance map.

1 Introduction

Barcode detection is required in a wide range of real-life applications. Computer vision algorithms vary considerably and each application has its own requirements. Many barcode localization methods have been developed for automatically segmenting barcode patterns from images.

The term barcode can be used for various types of visual codes. In this paper, we deal with classical 1D barcodes and stacked 2D barcodes. Barcodes are not human-readable and traditional devices have been widely adopted for personal use. The traditional barcode structure is simple: the variation of different thickness of parallel light and dark bars represent information. Such codes can be read optically by a machine. Code types vary from each other in what configurations of black and white bars correspond to a given character. The most frequent application of barcodes is the trade, e.g. in goods packing. The identification number visualized with symbols permit the use of electronic reading by machines which support and speed up the information streaming. Barcodes can have fix or variable length. In fix length codes, the standard specifies how many

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 464–475, 2013.

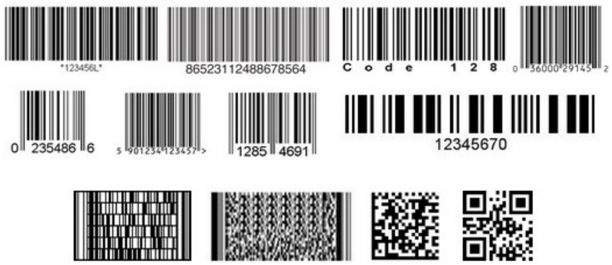


Fig. 1. Barcode patterns (from left to right). Top row (1D codes): Code39, Codabar, Code128, UPC-A; Middle row (1D codes): UPC-E, EAN-13, EAN-8, I2of5; Bottom row (2D codes): Codablock, PDF417, Data Matrix, QR.

characters are represented in a code, while the other type may encode an arbitrary number of characters. Different standard code types have specific features that help their localization. Figure 1 shows a set of standard 1D barcode types and some widely used 2D barcode types. [1]

Barcode localization methods have two competing objectives: detection accuracy and speed. For industrial environment, accuracy is crucial since undetected (missed) codes may lead to loss of profit. Processing speed is a secondary desired property of the detectors. On smartphones, accuracy is not so critical, since the device interacts with the user and reshooting is easily done, nevertheless, fast (and reasonably accurate) barcode detection is desirable. Various techniques are used to locate and decode barcodes from photographs: from the classical line scanning technique [2], through the widely studied morphological approaches [3–10], and recent studies using wavelets [11].

We propose a novel barcode detection algorithm based on bottom-hat filtering and other simple operations from mathematical morphology, which ensure high efficiency. We compare the effectiveness of the proposed method with several approaches from the literature and show that, in most cases, our algorithm outperforms the others. It also improves on the detection accuracy of our previous algorithm [10] for 2D stacked barcodes.

2 Methods

In this section, we present several barcode detection algorithms that use different approaches to determine the barcode location in an image. The methods presented in Sections 2.1, 2.2, and 2.3 were re-implemented according to the original papers. Where some details were not available in the papers, we made our best effort to fill in the gaps, and this is also noted in our description. The new proposed algorithm is presented in full detail in Section 2.4. For visual reference, Figure 2 illustrates intermediate stages of the processing by the described algorithms.

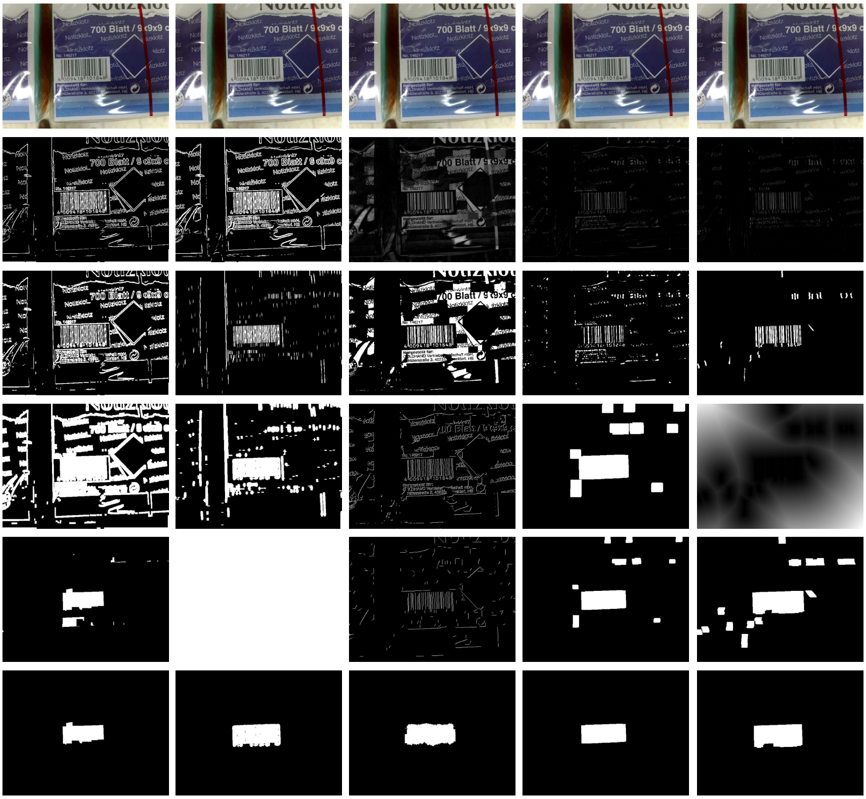


Fig. 2. Intermediate stages of the processing by the described algorithms. Columns: TT, ETJC, JJXQ, MKLN, and Proposed (see text). First row: original image, last row: final output.

Nowadays, the quality of digital images are usually very good, although low-quality recordings also exist. The reason for lesser quality may be e.g. the capturing device, and the environment may also impose problems. Therefore, often there is a need for correcting (improving) the image quality before the particular detection process takes place.

2.1 Method Based on Basic Morphological Operations

In Tuinstra's [3] algorithm, the author relies on that in the barcode region, the intensity difference between the stripes is high, so the gradient would highlight the bars. Sobel kernels are used to estimate the gradient in the x and y directions. Next, the gradient image is thresholded, pixels having a high gradient value are selected. On the binary image, first a hit-or-miss transformation is performed with a line structuring element, which, unfortunately, is not specified in the original article [3]. Next, morphological dilation is performed in order to merge nearby but not necessarily connected objects to be able to compose a region. The structuring element is square shaped, but its exact size was not fixed in the paper. We used a 10×10 block SE matching the size used for the previous step. In practice, the image size determines this parameter. Morphological erosion follows to discard thin objects from the image and remove undesired segments which were fused by the dilation. Here, the structuring element is greater than that used for dilation. We used a 20×20 block SE. The final operation is a solidity test that compares the number of pixels turned on in a region to the convex hull of the region. After this step, likely false positive objects are removed and only barcode regions remain.

2.2 Method Based on Image Scanning

The procedure of Telkin and Coughlan [4] was designed for visually impaired or blind people to facilitate their everyday lives. It is presented in two main phases. First, the noise level is reduced by Gaussian smoothing. Since the original paper did not specify the σ parameter, we used the same value of $\sigma = 0.3$ as in our Proposed method. Then, pixel gradient values are calculated using the Sobel operator, thus yielding a kind of edge enhancement in the image. Next, binarization is done so that pixels having a gradient above the threshold become white and the others black. We used a threshold value set to 95 % of the maximal gradient value. In the detection phase, first, the image is scanned in four directions (horizontally, vertically, and in direction of the diagonals (±45°)). The horizontal scan comes first, wherein those edge pixels are found whose orientation is vertical. The method looks in the vicinity of each edge pixel for opposite polarity pixels. An area will be kept if there is a sufficient number of such pixels in the area (as if they were part of a line segment). Vertical scan follows, when those segments remain which have almost the same beginning and end so they most likely belong to barcode areas. At the final stage of the detection the entropy value is calculated for each pixel in the resulting image that describes the disorder of intensities within a given neighborhood around each pixel.

2.3 Method Based on Bottom-Hat Filtering

The algorithm by Juett and Qui [5] is based on bottom-hat filtering. In preprocessing, the method corrects the non-ideal image with simple contrast stretching in order to highlight differences between light and dark areas. Next bottom-hat filtering is applied, wherein the size of the structuring element depends on the widest bar in the barcode to be detected. The article specified 25×25 block SE for images of size 720×480 pixels. After binarization, the contour is defined. It is followed by the step to determine the orientation of the bars, which is performed by directional image openings using a relatively large linear structuring element. These openings are performed at 16 different orientations, with a step of 11.25°. The directional opening images are summed and a low resolution density image is calculated, which is then converted back into binary. Each region represents a potential barcode region. In the last phase, objects whose area is smaller than a given threshold are eliminated.

2.4 Proposed Method Using Bottom-Hat Filtering

Our aim was to devise an algorithm with sufficiently high speed and accuracy. The presented algorithm is a modified version of our previously published method [10], and is summarized in Figure 3. Similarly to the above described methods [3–5], our algorithm consists of two main phases. In the preprocessing phase, the input image is converted to grayscale, because the pattern of dark bars on a light background is apparent regardless of some codes being printed using colors in certain applications.

```
ALGORITHM:
  $f := (f \bullet SE_1) - f$
  for all pixels do
      if $f_{i,j} > i_t$ then $f_{i,j} \leftarrow 1$
      else $f_{i,j} \leftarrow 0$
  Calculate Euclidean distance map
  Compute the minimum $X$ of row averages in the distance map
  Drop those pixels, whose distance value is greater than $X$
  $f := f \oplus SE_2$
  $f := f \ominus SE_3$
  for all components do
      if   component area $> min_a$ then  Record this component as a barcode segment
      else Discard this component
```

Fig. 3. Pseudocode of the Proposed algorithm

To reduce the image noise, we smooth with a Gaussian kernel using $\sigma = 0.3$. Edge enhancement follows. Instead of calculating grafient values (like done in most barcode localization methods), we use bottom-hat filtering that is also based on intensity differences. Although bottom-hat filtering is less attractive regarding operation time than other non-directional edge enhancement operations, its accuracy is higher. The proposed algorithm calculates the frequency of the most frequently occuring element. Later we refer to this value as $MaxFreq$. We defined a linear structuring element and its rotated version by 90°. We perform bottom-hat filtering using both structuring elements and later use the one that produces stronger matches. The grayscale images are then converted to binary, using a standard thresholding technique. We used $MaxFreq$ value and image size for calculating this threshold value. We assumed that, if $MaxFreq$ is very high, the image is noisy or blurred or both. In this case, we set the threshold to 95 % of the maximal value, otherwise the threshold is set to 80 % of the maximal value.

After preprocessing, the digital image is suited to finding segments which contain barcode. The resulting image so far contains many false positive regions. There are many small connected components which satisfy the criteria but do not belong to barcode regions. These false regions are eliminated using an area threshold. Selecting a too high threshold value could remove small barcodes from high-resolution images. If the image size is higher than 800×800, we set the threshold to half the size of the largest component, otherwise the threshold is set to 1/4 of the size of the largest component.

We take advantage of the structure of the barcode, the fact that it consists of approximately regularly spaced parallel stripes. So, for each pixel, we calculate the Euclidean distance of the pixel from the nearest nonzero pixel. Using the distance map, objects that are far from other objects can be easily dropped, and only nearby objects (sort of a cluster of bar segments) will be kept. The distance threshold is selected as the minimum of the row averages of the distance map.

Our observations showed, that for the final detection steps, mostly dense text regions remain along with the supposedly barcode areas. Since a barcode consists of a sequence of parallel bars that are located at varying distance from each other, they do not compose a connected component. Therefore, to merge these patterns, we use dilation with a square structuring element whose size is defined as $S = \max(40, \text{width of the widest bar} * 3)$. Nevertheless, this dilation may also thicken and merge unwanted, non-barcode locations as well. To cope with this problem, we also use the dual operation, i.e. erosion. Here, the structuring element is linear, consistent with the size of the 1D barcodes. The structuring element size is less than (about 1/3 of) the one used for dilation, i.e., it matches the width of the widest bar. After erosion, those areas that likely contain barcodes can be found. Of course, there may still be false positive objects. These are removed in the last step on the basis of their size and proportions. Similarly to the previous step, if the image size is higher than 800×800, objects smaller than half the size of the largest object are removed, otherwise objects smaller than 1/4 of the size of the largest object are removed.

3 Evaluation

In this section, we compare the discussed methods' effectiveness under specific conditions. We use the following acronyms for referring to the algorithms: TT (Timothy R. Tuinstra's method [3]), ETJC (Ender Telkin és James M. Coughlan's method [4]), JJXQ (James Juett és Xiaojun's method [5]), MKLN (our previous method [10]), and Proposed (Prop.) for our new proposed algorithm.

3.1 Test Suite, Test Environment, and Implementation

We generated barcodes digitally with the types shown in Figure 1. Only one base image was chosen for each code type. Test images contained one or three barcodes from each types, respectively, and images were affected by distortions. For each base image, we generated all combinations of the following properties:

rotation in every 15° from 0° to 180°, Gaussian blur filter using a 3×3 kernel with 6 different values of σ, additive noise from 0 % to 50 % with a step of 10 %, and an optional 15 % shear. In summary, the test set contained images presenting either 1 or 3 codes of 10 different barcode types, with 12 orientations, 6 different blur filters, 6 different rates of additive noise, with or without shear, totaling in 17,280 images. Figure 4 shows a selection of generated test images with various code content and applied distortions.

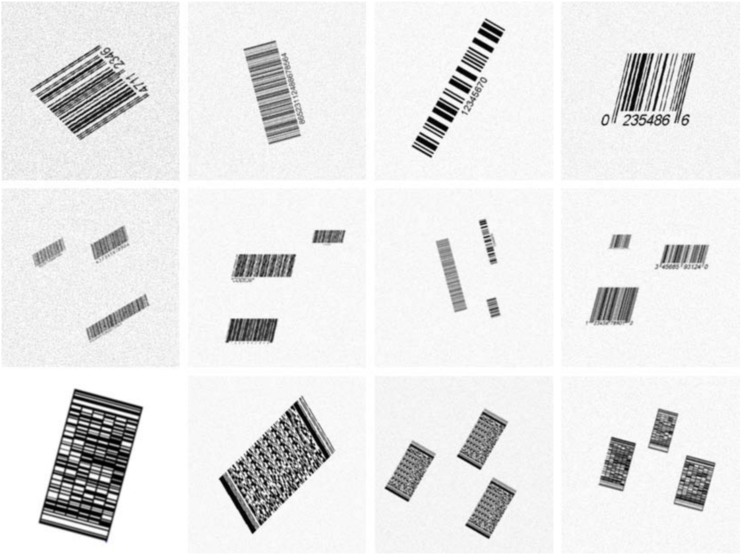


Fig. 4. A selection of generated test images with various code content and applied distortions

Another 100 images containing barcodes were collected from real-life images without any modifications. These images presented scratches, blur, minor light reflections and distortions also.

The methods were implemented in MATLAB using the Image Processing Toolbox. Evaulation was performed on a computer with Pentium(R) Dual-Core 2.30 GHz CPU.

3.2 Results and Discussion

In this section, we show how effective the implemented algorithms are on images with various characteristics, and compare their running time also. For calculating accuracy we used the Jaccard coefficient of similarity, measuring the overlap of the bounding boxes of the real and the detected barcode region. $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$, where A denotes the bounding box of the real barcode and B that of the detected code. This not only measures the success of detection, but also considers the delination accuracy of the methods.

The performance of the algorithms do not seem to depend on the applied stretching in the test images and also seem to be insensitive to rotation as well.

We can conclude that all five algorithms are capable of effectively localizing barcodes. ETJC method is fast, but it looses efficiency when there are several barcodes in the image and noise is also present. TT algotithm is slower than the others, which can most probably be attributed to that here the image is scanned in four different orientations. JJXQ, the method based on bottom-hat filtering falls off to the discussed fast algorithms with respect to running time. The process is slower than the others because in the detection phase the image is scanned in 16 different orientations. However, this brings better accuracy to the process at the expense of more computation.

Our proposed algorithms are also very fast. In most cases, faster procedures locate barcodes less accurately or not at all, as if loss of accuracy were the price for the speed gain. However, in our case, speed is not coupled with a significant cost in accuracy. From the evaluation output we can conclude that the proposed procedure has second best running efficiency for all test images (only our previous version executes faster), and in many cases it also shows best accuracy. The weakness of the method appears when the images is very noisy or when there are such image areas which are similar to a barcode.

The efficiency of the methods, i.e. processing time is also an important aspect. Table 1 presents the execution time of the detection methods for images with 1 or 3 barcodes. Here, one can easily appreciate the significant differences between the different approaches.

Table 1. Running time of the algorithms. Mean values (± standard deviation) (expressed in seconds) for all test images of a certain class

		TT	ETJC	JJXQ	MKLN	Proposed
1 code	1D	0.23±0.13	0.67±0.22	0.81±0.43	0.13±0.10	0.16±0.08
	2D	0.21±0.14	0.77±0.47	0.73±0.44	0.12±0.12	0.14±0.09
3 codes	1D	0.54±0.35	1.73±0.66	1.86±0.78	0.18±0.09	0.35±0.12
	2D	0.46±0.21	1.70±0.54	1.68±0.71	0.18±0.10	0.28±0.07

Subsequent tables show, how the algorithms behave on images which contain either one or three code pieces of various 1D and 2D barcodes types. The structure of barcode types varies, which also has an effect on how well the algorithms can perform. Accuracy of the detection methods for images containing a single or 3 pieces of codes for various code types is presented in Table 2. TT, ETJC, and JJXQ has very bad performance for Code128, however MKLN and the Proposed algorithm handle this variable-length code as well as the fixed-length types. For the 2D codes, ETJC shows exceptionally good accuracy, but the other methods also perform well on these stacked barcodes. There is more variance in performance on images with 3 codes, and some methods (TT, MKLN, Proposed)

Table 2. Accuracy of the algorithms for various types of codes, for images containing 1 or 3 codes. Mean values (expressed in percent) for all distorted test images with 1D barcodes and for all test images with 2D codes.

		1 code					3 codes				
		TT	ETJC	JJXQ	MKLN	Prop.	TT	ETJC	JJXQ	MKLN	Prop.
1D	Codabar	99.0	98.9	99.2	99.9	98.6	83.8	56.2	52.7	92.8	87.5
	Code128	50.4	63.6	70.9	95.7	93.7	83.6	95.0	79.6	80.1	84.8
	Code39	98.7	94.8	86.3	99.4	100.0	83.3	90.5	67.3	94.7	90.5
	EAN-13	98.8	94.7	91.7	98.9	99.3	83.3	91.4	76.6	99.3	99.2
	EAN-8	98.7	99.6	84.6	99.6	98.2	83.3	97.2	81.8	99.4	95.9
	I2of5	98.6	84.6	84.4	93.1	92.5	78.8	35.5	93.8	99.2	84.9
	UPC-A	99.1	98.2	94.9	86.4	100.0	74.6	65.5	92.6	99.9	95.4
	UPC-E	82.7	92.6	76.0	99.6	100.0	98.7	77.5	96.7	99.3	92.5
	all 1D	91.8	86.4	78.9	96.6	96.8	74.0	74.7	77.4	95.6	89.2
2D	Codablock	87.0	99.8	83.3	87.7	100.0	86.6	100.0	80.0	90.7	94.2
	PDF417	81.7	99.6	79.7	93.9	97.6	83.3	100.0	83.9	91.7	90.8
	all 2D	85.0	99.7	82.5	90.8	99.1	85.0	100.0	82.4	91.2	92.5

show a solid performance while the others very much depend on the type of code present in the image.

The accuracy of the algorithms degrades on distorted images. Next, we analyze the effect of distortions on efficiency. Table 3 shows the methods' behavior with respect to the level of blur applied to the test images. Image smoothing does not change the performance considerably. Although there is a noticeable inverse relation between values of σ and accuracy, this is not significant. It is interesting to note that in some cases the barcode detection accuracy is higher in images with three barcodes than for images with a single code present. This can be due to the fact that relative size (w.r.t. image size) of individual codes are smaller in the 3-code examples. ETJC, MKLN, and the Proposed method show outstanding accuracy in images which contain 2D barcodes and there are not considerable differences between the precision values.

In a similar manner, we analyzed how the algorithms perform on images having different levels of noise added. We can see from Table 4 that each algorithm is somewhat sensitive to noise. The studied literature methods loose much of their accuracy as the noise level increases, however, our methods only degrade to a lesser extent.

We also tested the methods on a set of 100 real-life images. These images contain 1D barcodes each. The accuracy of the algorithms are respectively 92 % (TT), 91 % (ETJC), 92 % (JJXQ), 94 % (MKLN), and 96 % (Proposed).

In Figure 5 we show two result images of our algorithm for 2D barcodes. Here, we have to locate all parts of the barcode, because stacked barcodes contain data everywhere and there is a much smaller visual redundancy as in 1D barcodes.

For the digitally generated images maximal size was 800×800. At this resolution TT and JJXQ cannot compete with the other two methods in speed. MKLN and the Proposed method is not far better on 800×800 images than ETJC, but

Table 3. Accuracy of the algorithms for different blur levels, for images containing a single code and for those containing 3 codes. Mean values (expressed in percent) for all distorted test images with 1D barcodes and for all test images with 2D codes.

		1 code					3 codes				
		TT	ETJC	JJXQ	MKLN	Prop.	TT	ETJC	JJXQ	MKLN	Prop.
1D	0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	0.5	91.2	88.4	91.3	96.1	95.7	93.8	94.4	92.4	98.0	100.0
	1.0	90.4	88.5	89.4	97.1	95.4	92.4	86.7	89.1	98.0	100.0
	1.5	90.4	88.5	89.4	96.8	95.4	91.7	86.7	89.1	97.0	100.0
	2.0	90.3	90.4	87.5	97.1	93.3	86.3	82.9	90.2	97.0	100.0
	2.5	90.4	90.4	87.5	97.1	93.3	82.7	82.0	90.2	97.0	100.0
	all 1D	90.5	89.1	89.6	96.7	94.3	82.7	77.9	90.6	97.5	100.0
2D	0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	0.5	88.8	100.0	83.1	95.8	100.0	89.1	100.0	83.2	100.0	100.0
	1.0	88.4	100.0	82.6	95.7	100.0	87.7	100.0	82.5	100.0	100.0
	1.5	87.7	100.0	82.6	95.7	100.0	87.0	100.0	82.5	100.0	100.0
	2.0	83.6	100.0	82.6	95.7	100.0	85.9	100.0	74.5	100.0	100.0
	2.5	83.6	100.0	82.6	95.7	100.0	85.0	100.0	74.5	100.0	100.0
	all 2D	87.5	100.0	83.3	95.8	100.0	87.0	100.0	82.9	100.0	100.0

Table 4. Accuracy of the algorithms for different noise levels, for images containing a single code and for those containing 3 codes. Mean values (expressed in percent) for all distorted test images with 1D barcodes (top), and for all test images with 2D codes (bottom).

		1 code					3 codes				
		TT	ETJC	JJXQ	MKLN	Prop.	TT	ETJC	JJXQ	MKLN	Prop.
1D	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	10	90.2	89.5	93.9	97.4	97.1	94.8	97.5	92.6	96.0	95.8
	20	89.7	86.9	87.5	97.4	97.1	89.4	87.6	89.5	96.3	91.6
	30	89.4	86.2	79.2	97.4	92.5	84.6	86.3	82.7	96.7	95.8
	40	90.1	90.1	78.5	97.8	92.5	87.2	84.9	79.9	96.7	95.8
	50	93.3	84.2	69.9	94.9	91.6	73.8	82.7	64.4	90.6	91.6
	all 1D	90.5	88.6	83.5	96.8	95.4	87.6	87.3	84.7	95.6	95.1
2D	0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	10	87.7	100.0	84.7	94.4	100.0	86.3	100.0	84.0	100.0	100.0
	20	87.7	98.6	81.9	94.2	100.0	85.8	100.0	78.9	100.0	100.0
	30	87.7	100.0	79.3	94.2	100.0	90.4	100.0	73.0	100.0	100.0
	40	85.7	100.0	76.7	94.2	100.0	86.3	100.0	72.6	100.0	100.0
	50	77.8	98.6	69.6	92.3	100.0	84.9	100.0	71.4	85.7	100.0
	all 2D	85.8	99.5	77.3	94.8	100.0	86.4	100.0	73.3	95.6	100.0

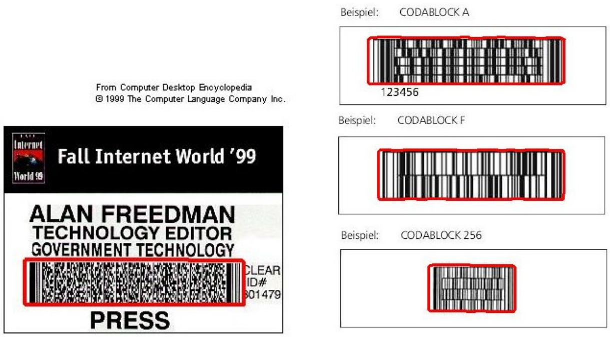


Fig. 5. Display examples of detection output by the propsed algorithm for PDF417 (left) and Codablock (right) code samples

when executing them on 2500×1722 images, ETJC runs for 3.2 sec, while our algorithms finish under 1.5 sec. The Proposed algorithm is somewhat slower than the MKLN version due to the extra computation of the distance map. We can state, that for larger images there are considerable execution time differences between the studied algorithms.

Our algorithms outperform the three algorithms from the literature both in terms of accuracy an speed. Note, that methods which use bottom-hat filtering in the preprocessing phase, have higher accuracy than the others, but their running times are quite different from each other. JJXQ looses its accuracy proportional to image degradation, while our proposed algorithms maintain higher accuracy for those cases as well. The tests also demonstrated that in all cases the methods' running time significantly grows with increasing image size. Among the two methods of ours (MKLN and Proposed), MKLN accuracy is slightly better than that of the Proposed version for the entire pool of our synthetic test images, including many 1D as well as 2D code examples. Regarding images with 2D stacked codes only, the newly proposed algorithm clearly outperforms the MKLN version.

4 Conclusion

We have presented a new version of our barcode detection algorithm and compared its performance (in terms of accuracy and speed) with our previous version and three other methods from the literature using similar ideas. We concluded that these algorithms do not specialize for individual barcode types, they can efficiently detect various types of 1D and stacked 2D barcodes, but there are subtle differences in performance. We have built a test database containing 17,280 synthetic images representing various degradations (blur, noise, shear) as well as collected 100 real images for evaluation. We demonstrated that the proposed new methods are less sensitive noise than those from the prior art. Our proposed methods outperform the other three in detecting 1D codes both in terms of accuracy and speed, and the new proposed version has better accuracy for 2D stacked barcode examples.

Acknowledgements. The work of the second author is supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

1. Palmer, R.C.: The Bar Code Book: Reading, Printing, and Specification of Bar Code Symbols. Helmers Pub. (1995)
2. Adelmann, R.: Toolkit for bar code recognition and resolving on camera phones - jump starting the internet of things. In: Informatik 2006 Workshop on Mobile and Embedded Interactive Systems (2006)
3. Tuinstra, T.R.: Reading Barcodes from Digital Imagery. PhD thesis, Cedarville University (2006)
4. Tekin, E., Coughlan, J.M.: An algorithm enabling blind users to find and read barcodes. In: Workshop on Applications of Computer Vision (WACV), pp. 1–8 (2009)
5. James Juett, X.Q.: Barcode localization using bottom-hat filter. NSF Research Experience for Undergraduates (2005)
6. Bodnár, P., Nyúl, L.G.: Efficient barcode detection with texture analysis. In: Proceedings of the Ninth IASTED International Conference on Signal Processing, Pattern Recognition, and Applications, pp. 51–57 (2012)
7. Bodnár, P., Nyúl, L.G.: Improving barcode detection with combination of simple detectors. In: The 8th International Conference on Signal Image Technology (SITIS 2012), pp. 300–306 (2012)
8. Chai, D.: Locating and decoding ean-13 barcodes from images captured by digital cameras. In: Fifth International Conference on Information, Communications and Signal Processing, pp. 1595–1599 (2005)
9. Lin, D.T., Lin, M.C., Huang, K.Y.: Real-time automatic recognition of omnidirectional multiple barcodes and dsp implementation. Machine Vision and Applications 22, 409–419 (2011)
10. Katona, M., Nyúl, L.G.: A novel method for accurate and efficient barcode detection with morphological operations. In: The 8th International Conference on Signal Image Technology (SITIS 2012), pp. 307–314 (2012)
11. Wachenfeld, S., Terlunen, S., Jiang, X.: Robust recognition of 1-d barcodes using camera phones. In: 19th International Conference on Pattern Recognition (ICPR 2008), pp. 1–4 (2008)

Faster Fuzzy Connectedness via Precomputation

Filip Malmberg and Robin Strand

Centre for Image Analysis, Uppsala University, Sweden
{filip,robin}@cb.uu.se

Abstract. We propose a method for accelerating the computation of fuzzy connectedness. The method is based on a precomputation step – the construction of a *supervertex* graph whose vertices are clusters of image elements. By constructing this supervertex graph in a specific way, we can perform the bulk of the fuzzy connectedness computations on this graph, rather than on the original image, while guaranteeing exact results. Typically, the number of nodes in the supervertex graph is much smaller than the number of elements in the image, and thus less computation is required. In an experiment, we demonstrate the ability of the proposed method to accelerate the computation of fuzzy connectedness considerably.

Keywords: Fuzzy Connectedness, Supervertex graph, Interactive segmentation

1 Background

It is often crucial that image processing methods are computationally efficient. This holds in particular for interactive image segmentation methods, where the feedback to the user should be computed in real-time. *Fuzzy connectedness*, introduced by Udupa et al. [6], forms the core of many methods for interactive image segmentation.

Here, we will present fuzzy connectedness in a graph theoretic framework. In this context, an image is represented by its *pixel adjacency graph*, i.e., a graph whose vertex set is the set of image elements, and whose edge set is given by an adjacency relation on the image elements. Each path through this graph is assigned a real valued *strength of connectedness.* The fuzzy connectedness between two image elements is defined as the maximum strength of connectedness of any path between the elements. Since its introduction, the fuzzy connectedness framework has been extended to support the use of vectorial data and prior information such as object/background intensities (through the homogeneity-based component of the fuzzy affinity), intensity distribution or local size (through the object feature-based component of the fuzzy affinity). See for example [1,2,8].

Here, we focus on the following computational problem:

Problem 1. *Given a subset of the elements in an image (called* seedpoints*), find a mapping that assigns to every image element the fuzzy connectedness between the element and the seedpoint to which it is most closely connected.*

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 476–483, 2013.

A formal definition of this problem is given in Section 3.2. Using dynamic programming, Problem 1 can be solved in $\mathcal{O}(n \log n)$ operations, where n is the number of image elements. While this is an efficient algorithm, the real-time requirement of fuzzy connectedness is not always met for large data sets. Here, we show that this computation can be accelerated by precomputation, while still guaranteeing exact results. The proposed approach is particularly effective if we want to solve Problem 1 many times for the same image, but with different sets of seedpoints, as is often the case in interactive segmentation.

The proposed method is based on pre-computing a *supervertex graph*, whose vertices are disjoint sets (*supervertices*) of image elements. We show that by constructing this supervertex graph in a specific way, we can guarantee the following property: If two image elements belong to different supervertices, then fuzzy connectedness between the pixels, in the pixel adjacency graph, is equal to the fuzzy connectedness between their corresponding supervertices. Typically, the supervertex graph has much fewer vertices than the original pixel adjacency graph, and thus less computations are required.

2 Related Work

Falcão et al. [3] proposed a *differential* algorithm for solving a generalized version of Problem 1. The differential algorithm solves the problem for a given set of seedpoints in sublinear time, provided that a solution is known for a *similar* set of seedpoints.

Another approach for accelerating the computation of fuzzy connectedness is given in [7], where a GPU implementation is presented. The authors report a speedup of about 7-14 times on large datasets. Note that this approach, due to the GPU programming, it is hardware dependent.

The proposed method is also related to the hierarchical clustering technique [5] in mathematical morphology, which is a sequence of non-overlapping clusters such that, at each level, each cluster is contained in a cluster in the next level. Common approaches include watershed-based, where the clustering is guided by watershed segmentations where adjacent regions are merged to get the hierarchical representation, and tree-based approaches, where flat-zones are iteratively merged with the most similar adjacent flat-zones.

3 Preliminaries

3.1 Graphs

We define a (undirected, edge weighted) *graph* as a triple $G = (V(G), E(G), A)$ where

- $V(G)$ is a set.
- $E(G)$ is a set of unordered pairs of distinct elements in V, i.e., $E \subseteq \{\{v, w\} \subseteq V \mid v \neq w\}$.
- A is a map $A : E \to [0, 1]$.

The elements of $V(G)$ are called *vertices* of G, and the elements of $E(G)$ are called *edges* of G. In order to simplify the notation, the vertices and edges of a graph will be denoted V and E instead of $V(G)$ and $E(G)$ whenever it is clear from the context which graph they belong to. An edge spanning two vertices v and w is denoted $e_{v,w}$. If $e_{v,w}$ is an edge in E, the vertices v and w are *adjacent*. The value of A represents the *affinity*, or degree of similarity, between adjacent vertices in the graph, i.e., $A(e_{v,w})$ is high if v and w are closely related.

A *path* in G is an ordered sequence of vertices $\pi = \langle v_i \rangle_{i=1}^{k} = \langle v_1, v_2, \ldots, v_k \rangle$ such that $e_{v_i,v_{i+1}} \in E$ for all $i \in [1, k-1]$. For any path $\pi = \langle v_1, v_2, \ldots, v_k \rangle$, we define the *reverse path* π^{-1} as $\langle v_k, v_{k-1}, \ldots, v_1 \rangle$. A path consisting of a single vertex is called a *trivial* path. We denote the origin v_1 and the destination v_k of π by $org(\pi)$ and $dst(\pi)$, respectively. If π and τ are paths such that $dst(\pi) = org(\tau)$, we denote by $\pi \cdot \tau$ the concatenation of the two paths. The set of all possible paths in G is denoted Π_G. The set of all possible paths in G starting at v and ending at w is denoted $\Pi_G(v, w)$. The subscript G will be omitted whenever it is clear from the context which graph we are referring to. In the following, we will only consider *connected* graphs, i.e., $\Pi(v, w) \neq \emptyset$ for all $v, w \in V$.

3.2 Fuzzy Connectedness

The *strength of connectedness* $\mu(\pi)$ of a path $\pi = \langle v_1, v_2, \ldots, v_k \rangle$ is defined as

$$\mu(\pi) = \begin{cases} 1 & \text{if } \pi \text{ is trivial;} \\ \min\limits_{i \in [1,k-1]} A(e_{v_i,v_{i+1}}) & \text{otherwise.} \end{cases} \quad . \tag{1}$$

Definition 1. *The* fuzzy connectedness*, $FC_G(v, w)$, between two vertices $v, w \in V$ is defined as*

$$FC_G(v, w) = \max_{\pi \in \Pi(v,w)} \mu(\pi) \ . \tag{2}$$

We are now ready to give a formal definition of Problem 1, that was outlined in Section 1.

Problem 1. *Given a set of vertices $S \subseteq V$ (called* seedpoints*), find a mapping $\phi : V \to [0, 1]$ such that $\phi(v) = \max\limits_{w \in S} FC_G(v, w)$.*

Problem 1 can be solved in $\mathcal{O}(|V| \log |V|)$ operations by Algorithm 1 (with $\lambda = 0$). The key to an efficient implementation of Algorithm 1 is to store the set Q in a data structure that allows rapid extraction of the element with maximum value, e.g., some kind of priority queue. See, e.g., [4] for details.

4 Supervertex Graphs

Let $G = (V, E, A)$ be a graph and let $C = \{C_1, C_2, \ldots, C_k\}$ be a partition of V. Every vertex $v \in V$ belongs to exactly one element of C, denoted $\mathcal{C}(v)$.

Algorithm 1. $A_1(G, S, \lambda)$

```
Input: A graph $G = (V, E, A)$ and a set $S \subseteq V$ of seedpoints. $\lambda \in [0, 1]$
Output: A map $\phi : V \to [0, 1]$.
Auxiliary: Two sets of vertices $\mathcal{F}$,$\mathcal{Q}$ whose union is $V$.
Set $\mathcal{F} \leftarrow S$, $\mathcal{Q} \leftarrow V \setminus S$;
For all $v \in \mathcal{Q}$, set $\phi(v) \leftarrow 0$;
For all $v \in \mathcal{F}$, set $\phi(v) \leftarrow 1$;
while $\mathcal{Q} \neq \emptyset$ and $\max_{v \in \mathcal{Q}} \phi(v) \geq \lambda$ do
    Remove from $\mathcal{Q}$ a vertex $v$ such that $\phi(v)$ is maximal, and add it to $\mathcal{F}$;
    foreach $w \in \mathcal{N}(v) \cap \mathcal{Q}$ do
        if $\min(\phi(v), A(e_{v,w})) > \phi(w)$ then
            Set $\phi(w) \leftarrow \min(\phi(v), A(e_{v,w}))$;
```

Definition 2. *The* boundary ∂C *of the partition* C *is the set* $\{e_{v,w} \in E(G) \mid \mathcal{C}(v) \neq \mathcal{C}(w)\}$.

Definition 3. *Let* $C_i, C_j \in C$. *The* boundary $\partial C_i, C_j$ *between* C_i *and* C_j *is the set* $\{e_{v,w} \in E(G) \mid v \in C_i$ *and* $w \in C_j\}$.

Definition 4. *The* supervertex graph $\mathcal{S}$ *induced by* G *and* C *is defined by:*

$$V(\mathcal{S}) = C \ . \tag{3}$$

$$E(\mathcal{S}) = \{\{C_i, C_j\} \mid \partial C_i, C_j \neq \emptyset\} \ . \tag{4}$$

$$A(e_{C_i,C_j}) = \max_{e \in \partial C_i, C_j} A(e) \textit{ for all } e_{C_i,C_j} \in E(\mathcal{S}) \ . \tag{5}$$

Definition 5. *Let* $W \subseteq V(G)$ *and let* $\mathcal{S}$ *be the supervertex graph induced by* G *and* C. *The* image *of* W *in* $\mathcal{S}$ *is defined as the set*

$$\{C_i \mid \exists v \in W \textit{ such that } v \in C_i\} \ . \tag{6}$$

Definition 6. *Let* $\pi = \langle v_i \rangle_{i=1}^{k} \in \Pi_G$ *and let* $\mathcal{S}$ *be the supervertex graph induced by* G *and* C. *The* image *of* π *in* $\mathcal{S}$ *is defined as the sequence obtained by removing from* $\langle \mathcal{C}(v_i) \rangle_{i=1}^{k}$ *all but one vertex from every set of consecutive equal vertices.*

5 α-Connectedness

Let $\alpha \in [0, 1]$ be a fixed value. We now define a binary relation, $\alpha-$*connectedness*, on V as follows

Definition 7. *Two vertices* $v, w \in V$ *are* α-connected *if* $FC(v, w) \geq \alpha$. *The notation* $v \underset{\alpha,G}{\sim} w$ *is used to indicate that* v *and* w *are* α*-connected on* G.

Theorem 1. *α-connectedness is an equivalence relation, i.e., for any $a, b, c \in V$ it holds that*

1. $a \underset{\alpha,G}{\sim} a$.
2. *if* $a \underset{\alpha,G}{\sim} b$ *then* $b \underset{\alpha,G}{\sim} a$.
3. *if* $a \underset{\alpha,G}{\sim} b$ *and* $b \underset{\alpha,G}{\sim} c$ *then* $a \underset{\alpha,G}{\sim} c$.

Proof. 1. $\mu(\langle a \rangle) = 1 \geq \alpha$ for any $\alpha \in [0, 1]$.
2. If $\pi \in \Pi_{a,b}$ such that $\mu(\pi) \geq \alpha$, then $\pi^{-1} \in \Pi_{b,a}$ and $\mu(\pi^{-1}) = \mu(\pi)$.
3. Let $\pi \in \pi_{max}(a, b)$ and let $\tau \in \pi_{max}(b, c)$. Then $\mu(\pi \cdot \tau) = max(\mu(\pi), \mu(\tau)) \geq \alpha$, and so $a \underset{\alpha,G}{\sim} c$.

□

Thus, the set of equivalence classes of $\underset{\alpha,G}{\sim}$ is a partition of V.

6 Method

Let G be a graph, let $S \subseteq V(G)$ be a set of seedpoints, let $\alpha \in [0, 1]$ be a fixed value, and let C be the set of equivalence classes of $\underset{\alpha,G}{\sim}$. Let $\mathcal{S}$ be the supervertex graph induced by G and C, and let S' be the image of the set S in $\mathcal{S}$. We propose the following procedure for solving Problem 1:

1. Compute $\phi_G = A_1(G, S, \alpha)$. Additionally, store the set $\mathcal{F}$ containing all vertices $v \in V(G)$ for which $\min_{w \in S} FC_G(v, w) \geq \alpha$.
2. Compute $\phi_{\mathcal{S}} = A_1(\mathcal{S}, S', 0)$.
3. The solution to Problem 1 is then given by

$$\min_{w \in S} FC_G(v, w) = \begin{cases} \phi_G(v) \text{ if } v \in \mathcal{F}; \\ \phi_{\mathcal{S}}(v) \text{ otherwise.} \end{cases} . \tag{7}$$

In the remainder of this section, we will prove the correctness of the above procedure.

Lemma 1. *For all $e_{v,w} \in \partial C$, it holds that $A(e_{v,w}) < \alpha$.*

Proof. Assume to the contrary that $A(e_{v,w}) \geq \alpha$. Then $v \underset{\alpha,G}{\sim} w$, contradicting that $e_{v,w} \in \partial C$.

Lemma 2. *Let $\pi = \langle C_i \rangle_{i=1}^{k}$ be a non-trivial path in $\mathcal{S}$, and let ϖ be the set of paths in G whose image in $\mathcal{S}$ is π. Then $\mu(\pi) = \max_{\tau \in \varpi} \mu(\tau)$.*

Proof. Let $\tau \in \varpi$. We can rewrite τ on the form $\tau_1 \cdot \tau_2 \cdot \ldots \cdot \tau_k$, where each subpath τ_i is contained in C_i. Then, by the definition of C, there exists a path $\tau' = \tau'_1 \cdot \tau'_2 \cdot \ldots \cdot \tau'_k \in \varpi$ where each subpath τ'_i satisfies the following properties:

1. $org(\tau'_i) = org(\tau_i)$.
2. $dst(\tau'_i) = dst(\tau_i)$.
3. $\mu(\tau'_i) \geq \alpha$.

From Lemma 1 it follows that $\mu(\tau') = \max\limits_{i \in [1..k-1]} A(e_{dst(\tau'_i), org(\tau'_{i+1})})$. The maximum value of $\mu(\tau')$ is thus attained by selecting the endpoints of each subpath τ'_i such that $e_{dst(\tau'_i), org(\tau'_{i+1})} \in \operatorname*{argmax}\limits_{e \in \partial C_i, C_{i+1}} A(e)$. The strength of connectedness of this optimal path is $\mu(\tau') = \min\limits_{i \in [1..k]} \max\limits_{e \in \partial C_i, C_{i+1}} A(e) = \min\limits_{i \in [1..k]} A(e_{C_i, C_{i+1}}) = \mu(\pi)$.

□

We are now ready to derive our main result:

Theorem 2. *Let $v, w \in V(G)$, such that $FC_G(v, w) < \alpha$. Then $FC_G(v, w) = FC_{\mathcal{S}}(\mathcal{C}(v), \mathcal{C}(w))$.*

Proof. For all $\pi \in \Pi_{\mathcal{S}}$, let $\varpi(\pi)$ be the set of paths in G whose image in $\mathcal{S}$ is π. Then $\Pi_G(v, w) = \bigcup_{\pi \in \Pi_{\mathcal{S}}(\mathcal{C}(v), \mathcal{C}(w))} \varpi(\pi)$, and $FC_G(v, w) = \max\limits_{\tau \in \Pi_G(v,w)} \mu(\tau) = \max\limits_{\pi \in \Pi_{\mathcal{S}}(\mathcal{C}(v), \mathcal{C}(w))} \max\limits_{\tau \in \varpi(\pi)} \mu(\tau)$. Since $FC_G(v, w) < \alpha$, the image of any path from v to w in $\mathcal{S}$ is non-trivial, and so by Lemma 2 we have $FC_G(v, w) = \max\limits_{\tau \in \Pi_G(v,w)} \mu(\tau) = \max\limits_{\pi \in \Pi_{\mathcal{S}}(\mathcal{C}(v), \mathcal{C}(w))} \mu(\pi) = FC_{\mathcal{S}}(\mathcal{C}(v), \mathcal{C}(w))$. □

7 Experiment

To demonstrate the performance of the proposed method, we performed an experiment using the two images shown in Figure 1. Pixel adjacency graphs were built from these images using standard 4-connectivity. The affinity between adjacent pixels was defined as

$$A(e_{v,w}) = 1 - \frac{1}{3}(|I_R(v) - I_R(w)| + |I_G(v) - I_G(w)| + |I_B(v) - I_B(w)|) \ , \quad (8)$$

where $I_R(v)$, $I_G(v)$ and $I_B(v)$ are the intensities of the red, green and blue channels, respectively, at the image element corresponding to the vertex v.

Fifty sets of seedpoints, each containing a single randomly selected seedpoint, were created. For each set of seedpoints, Problem 1 was solved using the proposed method with α-values ranging from 0 to 1. For reference, we also solved the problem using Algorithm 1 with $\lambda = 0$. The total execution time was measured for both methods. Figure 2 shows the relative execution time of the proposed method, compared to running Algorithm 1 without the precomputation step. The reduction in execution time depends on the value of α. For the "Peppers" image, the maximum measured speedup was about a factor 4. For the "Squares" image the maximum measured speedup was about a factor 40.

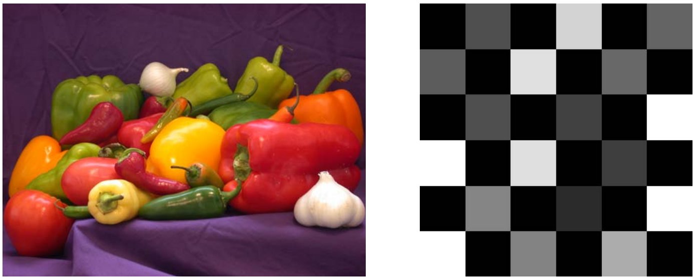


Fig. 1. Images used in the experiment. (Left) "Peppers". (Right) "Squares"

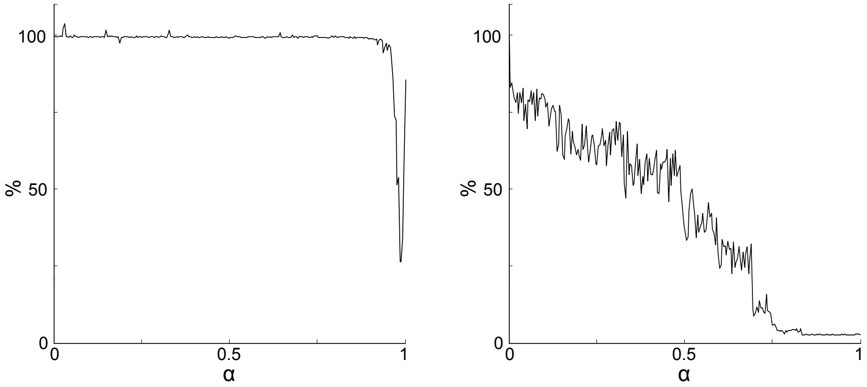


Fig. 2. Relative execution time. (Left) Execution time for the "Peppers" image. (Right) Execution time for the "Squares" image.

8 Conclusions

We have presented a method for accelerating the computation of fuzzy connectedness on undirected graphs. The method requires a precomputation step – the construction of a supervertex graph – and it is therefore most effective in situations where we wish to repeatedly compute fuzzy connectedness on a static graph with a varying set of seedpoints. One application where this situation frequently occurs is interactive image segmentation.

In an experiment, we have demonstrated the ability of the proposed method to accelerate the computation of fuzzy connectedness considerably. The reduction in computation time depends on a parameter α used in the construction of the supervertex graph. In future work we intend to investigate if this parameter can be selected automatically, based on, e.g., the histogram of edge affinities in the graph.

References

1. Ciesielski, K.C., Udupa, J.K.: Affinity functions in fuzzy connectedness based image segmentation i: Equivalence of affinities. Computer Vision and Image Understanding 114(1), 146–154 (2010)
2. Ciesielski, K.C., Udupa, J.K.: Affinity functions in fuzzy connectedness based image segmentation ii: Defining and recognizing truly novel affinities. Computer Vision and Image Understanding 114(1), 155–166 (2010)
3. Falcão, A.X., Bergo, F.P.G.: Interactive volume segmentation with differential image foresting transforms. IEEE MI 23(9), 1100–1108 (2004)
4. Nyúl, L.G., Falcão, A.X., Udupa, J.K.: Fuzzy-connected 3D image segmentation at interactive speeds. Graph. Models 64(5), 259–281 (2002)
5. Soille, P., Najman, L.: On morphological hierarchical representations for image processing and spatial data clustering. In: Köthe, U., Montanvert, A., Soille, P. (eds.) WADGMM 2010. LNCS, vol. 7346, pp. 43–67. Springer, Heidelberg (2012)
6. Udupa, J.K., Samarasekera, S.: Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. Graphical Models and Image Processing 58(3), 246–261 (1996)
7. Zhuge, Y., Cao, Y., Miller, R.W.: GPU accelerated fuzzy connected image segmentation by using CUDA. In: Engineering in Medicine and Biology Society, EMBC 2009. Annual International Conference of the IEEE, pp. 6341–6344 (2009)
8. Zhuge, Y., Udupa, J.K., Saha, P.K.: Vectorial scale-based fuzzy-connected image segmentation. Computer Vision and Image Understanding 101(3), 177–193 (2006)

Mask Connectivity by Viscous Closings: Linking Merging Galaxies without Merging Double Stars

Ugo Moschini[1], Scott C. Trager[2], and Michael H.F. Wilkinson[1]

[1] Johann Bernoulli Institute, University of Groningen, Groningen, The Netherlands
[2] Kapteyn Astronomical Institute, University of Groningen, Groningen, The Netherlands

Abstract. Second-generation connectivity opened the path to the use of mask images to freely define connectivity among the image components. In theory, any image could be treated as a mask image that defines a certain connectivity. This creates a new problem in terms of which image to use. In this paper, clustering masks suitable for the analysis of astronomical images are discussed. The connectivity defined by such masks must be capable of preserving faint structures like the filaments that link merging galaxies while separating neighboring stars. In this way, the actual morphology of the objects of interest is kept. This is useful for proper segmentation. We show that viscous mathematical morphology operators have a superior performance and create appropriate connectivity masks that can deal with the characteristic features of astronomical images.

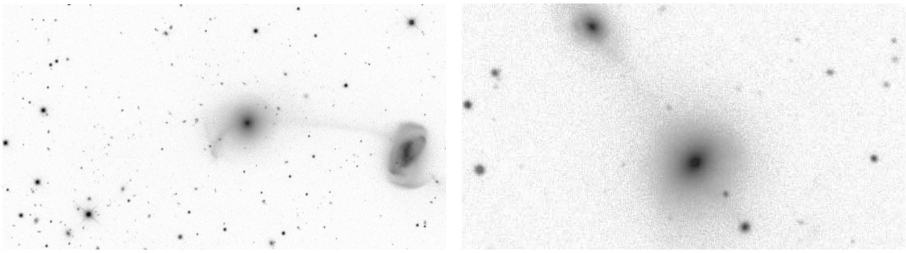
1 Introduction

Second-generation connectivity [1–3] is an effective generalization of the notion of connectivity. It models directly the perceptual clustering or partitioning that is automatically made by an human observer while analysing an image or a scene. Clustering lets objects close to each other in morphological terms to be considered as a single entity, whereas partitioning can split objects that are barely connected. Algorithms that can compute attribute filters using second-generation connectivity based on Max-Trees have been already presented, providing both sequential and parallel implementations [3, 4] for shared-memory machines. Thus, second-generation connectivity is an enabling technology in real applications. In principle, any image can act as a mask that defines a connectivity, independently from how it is computed. Given this lack of restriction, the problem is to evaluate which masks work better according to the task at hand.

An important task in astronomy is to distinguish stars from galaxies and then to classify the way in which the galaxies interact. It would be desirable to achieve a clear separation of those objects that an human observer would immediately identify as separate entities. Fig. 1 shows two examples of merging galaxies connected by a filament. In such examples, nearby or double stars are visually interpreted as separate stars while merging galaxies connected by a filament

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 484–495, 2013.

are identified as two linked objects. The state-of-the-art software used in the astronomical community used for the detection, segmentation, and classification of objects in astronomical images is called SExtractor [5]. SExtractor is based on an estimation of the background grey levels to establish a threshold above which an object is considered as such. Thresholding often removes not only noise in the background but also important, if faint, structures like the filaments between interacting galaxies. SExtractor would benefit from second-generation connectivity, exploiting the notion of connected components of an image rather than just cutting at a certain grey level threshold. The focus of this paper is on finding connectivity masks that can be used specifically in the analysis of astronomical images, moving from standard to viscous mathematical morphology operators to generate them.



(a) (b)

Fig. 1. Filaments in merging galaxies; images taken from the Sloan Digital Sky Survey.

Filaments connecting merging galaxies such as those in Figure 1 reveal information about the interaction between two galaxies. A filament is made of bright speckles and faint elongated structures. These components must be treated as a single connected entity and they should not be split up in smaller components, which individually cannot describe the structure of the filament or its presence. Clustering together the components of a filament, considering it a connected component, can be achieved with connectivity masks generated by extensive operators. The main problem is that, while we want to group together faint and diffuse structures broken up by noise, we want to keep closely-spaced bright structures, which have a far better signal-to-noise ratio, separated. We need operators which adapt the degree of clustering to the local image intensity. Viscous operators [6–8] seem ideally suited for the job.

The paper starts with the explanation of second-generation connectivity. Standard and viscous morphological operators are then presented. The next sections are dedicated to the comparison of suitable masks for astronomical purposes, showing how viscous morphological operators help in generating better masks. We finish with conclusions and a discussion of future work.

2 Second-Generation Connectivity

Let a binary image X be represented as a subset of some universal set V. Connected filtering operates on the partition of V generated by the *connected components* of X. In practice, V is often the discrete grid $\mathbb{Z}^n$, with $n = 2$ as the most common case. Given this grid-based representation of the image domain, it is natural to use adjacency relations to define connectivity. Therefore, in image processing the standard form of connectivity is graph-based path connectivity, in which a set is called connected if there exists a path made of adjacent (foreground) points within the set that connects every pair of its points. To facilitate this, we can extend our representation of a binary image as a subset of V to that of a subgraph of the undirected graph (V, E), with $E \subseteq V \times V$ the set of edges.

A connected component is a connected set of maximal extent in the image. More explicitly, a connected component to which some point $x \in X$ belongs is the union of all the connected sets $C \subseteq X$ containing element x. Connected components are retrieved by a family of connected openings denoted as $\Gamma_x(X)$, $x \in V$.

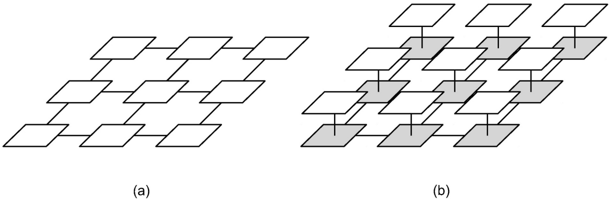


Fig. 2. Mask connectivity seen as modified path connection in a graph: (a) regular 4-connectivity; (b) mask connectivity. White nodes are original pixels, grey nodes mask pixels.

The concept of second-generation connectivity requires the creation of *connectivity masks* that define a connectivity over an image. The easiest way to visualize this is that we change the graph on which we work from (V, E) to (V', E'), as shown in Figure 2. We extend the set of vertices of the graph by adding a separate plane containing the mask image, and thus $V' = V \cup M$, where M is the set of mask-image pixels. We also remove the edges linking the original pixels together. The mask image pixels (shown in grey) are connected to each other in a conventional way. Furthermore, each original pixel is linked by an edge to the corresponding mask pixel. This means the set of edges $E' \subseteq M \times (V \cup M)$ and is disjoint from the original set of edges E. Thus, a pair of pixels x and y in the original image is connected at grey level h if there is a path $\{x_0, x_1, \ldots, x_n\} \subset M$ with $(x_0, x), (x_n, y) \in E'$ and such that $f(x_i) \geq h$, for all x_i. Thus, the paths through the mask dictate the connectivity. When computing any properties of the connected components, the mask pixels are ignored, and only the original pixels contribute [9].

There is a simple relationship between the standard connectivity opening Γ_x and the mask-based one Γ_x^M,

$$\Gamma_x^M(X) = \begin{cases} \Gamma_x(M) \cap X & \text{if } x \in X \cap M \quad (1a) \\ \{x\} & \text{if } x \in X \setminus M \quad (1b) \\ \emptyset & \text{otherwise} \quad (1c) \end{cases}$$

where M is the binary connectivity mask. A connectivity mask suitable for clustering can be created through any desired extensive operator. For a given mask M, the second-generation connected opening defined by $\Gamma_x^M(X)$, with X the image domain, returns as connected sets the intersection of the connected components in the mask image with the image itself. Therefore, the components in X are considered connected on the basis of the connectivity in the mask M rather than of the connectivity in X.

Using this approach, the topological aspects of the components of the mask image are exploited: the underlying connectivity is based on those and at the same time the shape of the original components is preserved. Components that are not connected in the original image can be connected through the underlying mask. For example, if the mask is computed from a dilation of the original, the concept of perceptual groups of image objects are perceived as clusters of connected components if the distance between them is lower than a given threshold, defined by the size of the structuring element. An example of the effect of a clustering mask is shown in Fig. 3 for the globular cluster Messier 13.

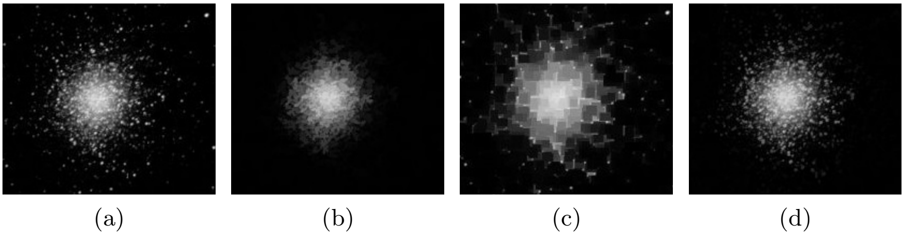


Fig. 3. Separating globular custer Messier 13 from isolated stars by an area thinning preserving connected components with area between 50 pixels and a quarter of the image size: (a) the original image; (b) the filtered image using no clustering mask; (c) the mask generated through a standard closing; (d) the filtered image using (c) as a connectivity mask.

3 Standard and Viscous Operators

The goal of viscous mathematical morphology [6–8] is to adapt the activity of a structuring element to the local intensity information of an image. Morphological filtering based on standard openings and closing applies the same transformation for all the image level sets. It is desirable to have different smoothing for different grey values, for example in situations in which it is known that components with high grey value do not require the same smoothing as components with lower grey value. A gray level image f can be thresholded into level sets $X_h = \{x \in E : f(x) \geq h\}$ where h represents the intensity value, ranging from the minimum to the maximum possible intensity value. The image can be reconstructed from all the threshold sets by taking their supremum $\bigvee_h h \cdot \chi_h(f)$. The *characteristic function* χ_h equals 1 if $x \in X_h$, else $\chi_h(x) = 0$. Standard dilation δ_λ and erosion ϵ_λ of an image f by a structuring element of size λ process all the threshold sets with the same structuring element and commute with grey scale thresholding:

$$\delta_\lambda(f) = \bigvee_h h\delta_\lambda[\chi_h(f)] \qquad \epsilon_\lambda(f) = \bigvee_h h\epsilon_\lambda[\chi_h(f)] \tag{2}$$

In viscous mathematical morphology, the size of the structuring element is not a fixed value λ. It is a function $\lambda(h)$ defined for every intensity value h:

$$\delta_\lambda^{visc}(f) = \bigvee_h h\delta_{\lambda(h)}[\chi_h(f)] \qquad \epsilon_\lambda^{visc}(f) = \bigvee_h h\epsilon_{\lambda(h)}[\chi_h(f)] \tag{3}$$

The function $\lambda(h)$ has been studied in two variants in previous work from Vachier and Meyer [7]. They define two functions to model two different situations. In the first case, regions of low intensity are dilated while points of higher intensity are left unchanged, and vice-versa in the second case. The function proposed for the former case was $\lambda(h) = \lambda_m(M - h)/M$ and $\lambda(h) = \lambda_m h/M$ in the latter, where λ_m is the maximum possible size of the structuring element and $[0, M]$ is the intensity range of the image. For the purposes of our work we are interested in the first case that let faint objects be dilated more, thus increasing the likelihood of identifying filaments. The equation that has been used in our work is $\Lambda(h)$, for $h \geq 1$:

$$\Lambda(h) = \lambda_m \frac{1}{h^p} \tag{4}$$

Figure 4 illustrates $\Lambda(h)$ for various p. Experiments were performed with different values for the parameter p, ranging in the interval from 0.2 to 0.5 in which good results were obtained. Since equation 4 must ideally create the biggest dilation for the lowest intensity value actually present in the image, the equation is rewritten as $\Lambda(h) = \lambda_m \frac{1}{(h-\min(f))^p}$, where $\min(f)$ is the lowest intensity value in the input image f, as a simple bias correction.

The dilation δ_Λ and erosion ϵ_Λ derived from equation 4 define an adjunction and produce openings γ^Λ and closings ϕ^Λ:

$$\delta_\Lambda(f) = \bigvee_h h\delta_\Lambda[\chi_h(f)] \qquad \epsilon_\Lambda(f) = \bigvee_h h\epsilon_\Lambda[\chi_h(f)] \tag{5}$$

and

$$\gamma^\Lambda = \bigvee_h h \cdot \gamma_\Lambda \chi_h \qquad \phi^\Lambda = \bigvee_h h \cdot \phi_\Lambda \chi_h \tag{6}$$

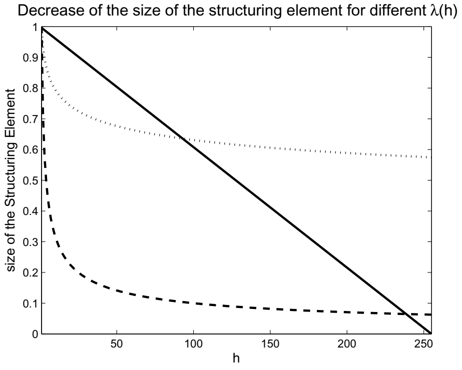


Fig. 4. Structuring element radius as a function of grey level h. Solid line shows the equation studied by Vachier and Meyer; dotted and dashed lines show equations $\Lambda(h)$, with $p = 0.1$ and $p = 0.5$ respectively.

where $\gamma_j = \delta_j \epsilon_j$ and $\phi_j = \epsilon_j \delta_j$ are the standard opening and closing by a structuring element of size j. The closing ϕ^{Λ} can be decomposed in the corresponding dilation and erosion:

$$\delta_{\Lambda}^{visc}(f) = \bigvee_{h} h\delta_{\Lambda(h)}[\chi_h(f)] \qquad \epsilon_{\Lambda}^{visc}(f) = \bigvee_{h} h\epsilon_{\Lambda(h)}[\chi_h(f)]. \tag{7}$$

This means that a connectivity mask suitable for clustering can be obtained using a viscous closing implemented as $\phi^{\Lambda} = \epsilon_{\Lambda}^{visc}\delta_{\Lambda}^{visc}$, allowing for the preservation of high intensities while dilating lower intensities.

4 Experiments and Discussion

The experiments aim to evaluate viscous operators in the generation of connectivity masks suitable to be used for astronomical images. To identify a faint, granular, galactic filament, clustering with a fairly large structuring element is required at low grey levels. On the other hand, bright objects like stars do not need to be processed by any underlying connectivity mask since they should be identified as separate objects: little or no clustering is required in this case.

Reconstruction from markers [10] is used here to determine at which level two objects are clustered together, being that a sign that any structure linking the two objects is preserved through the mask. A measure in percentage that quantifies the clustering will be given. We placed a δ-function marker in the core of one of the two galaxies undergoing merger, or alternatively, in one of two closely-spaced stars. The basic idea is shown in Figure 5. If no mask is used, reconstruction restores the adjacent source at the level of the saddle point between the two sources. When a mask is obtained from an extensive operator, it may restore more of the second source, with a maximum at the lower of the two peak levels.

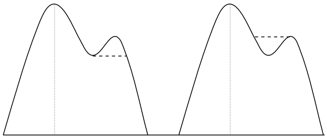


Fig. 5. Measuring level of connectivity in two closely-spaced sources. The solid curves indicate brightness profiles of two sources. The vertical fine dotted lines indicate the δ-function markers. On the left, the thick dashed line indicates the reconstruction from the marker using regular connectivity. On the right, the thick dashed line shows the worst-case merging height for clustering connections

4.1 Evaluation of the Experiments

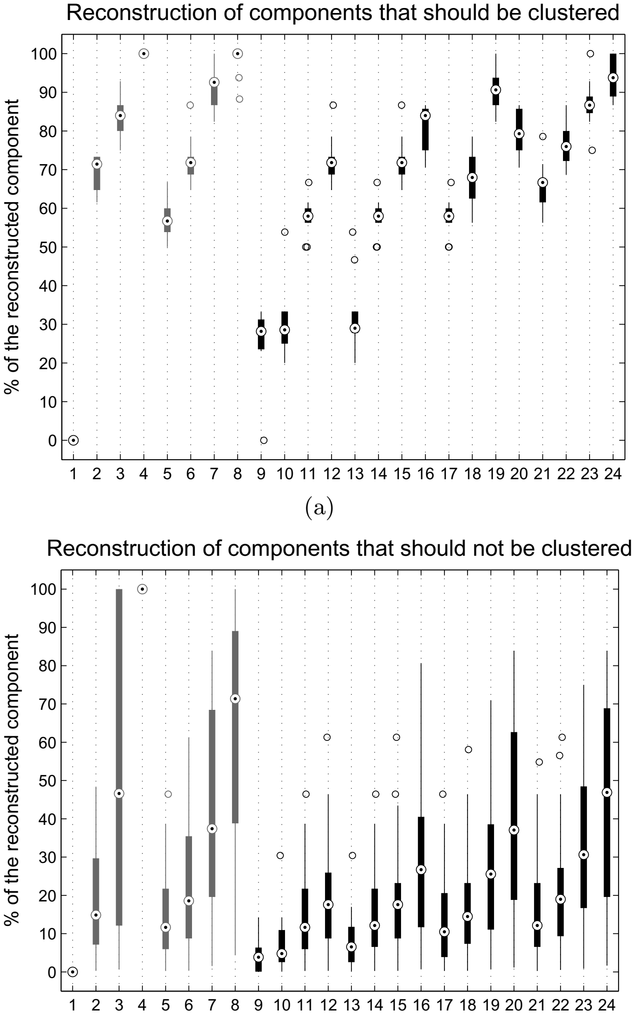
A percentage value is used to quantify how much the adjacent source is restored above the saddle point. A zero percent value is assigned to the intensity value of the adjacent component after a reconstruction in absence of clustering. It is not possible to reconstruct such a component at an intensity lower than the intensity value of the saddle point. A one hundred percent value is assigned to the intensity value of the adjacent component after a reconstruction that used a mask obtained from a standard dilation with an S.E. diameter of 20 pixels, corresponding to the maximum intensity that can be restored. Experiments were conducted varying the diameter of the structuring element used for the creation of the masks both for standard and viscous operators. The structuring element used in all the morphological operations has a circular shape, with a diameter ranging from 5 to 30 pixels. Viscous masks have been produced changing also the value of the parameter p that shapes the rate of decrease of the diameter length of the structuring element. The mask images were generated by standard dilations, closings and viscous closings. Ten 16-bit integer images taken in the R-band images of merging galaxies connected by filaments were selected from the Sloan Digital Sky Survey Data Release 7 [11] archives. Code was implemented to deal with 16-bit images. From the equations 3, a naive implementation of a viscous mask is obtained by dilating or eroding all the level sets and taking the supremum. To avoid looping over all the level sets in the intensity range, our implementation works by looping instead over all the possible different sizes of the structuring element. We use an adaptation of the multi-erode algorithm from [12] to perform this efficiently. The time complexity scales now with the number of possible different diameters that the structuring element can have. Hence, the code would work also on floating point images where the number of level sets is in principle infinite. Wall-clock time needed to generate a mask reduces by two orders of magnitude. For the max-tree computation the algorithm for high dynamic range images from [13] was used.

In every image, three positions for the markers were identified. A first marker was placed on the rightmost galaxy between the two connected by a filament in every image, the other two markers were placed in locations where two objects close to each other but not interacting were identified; either two stellar objects, or two galaxies not showing signs of any interaction between them. The percentage of the reconstructed adjacent component quantifies how much these objects are considered bound together. The box diagrams in Figure 6 summarise the experiments. Figure 6a shows how the masks perform in clustering desired objects together, while Figure 6b shows how they perform in keeping separate the objects that do not interact. The numbers on the x-axis indicate the twenty-four connectivities that have been tested as listed in Table 1. Index 1 shows the case where no mask image is used to define an underlying connectivity: in fact, it shows a value of zero percent meaning no clustering above the level of the saddle point. The mask generated by a standard dilation with a diameter of 20 pixels is assigned to the index 4 and it has value of one hundred per cent. Parameter p used in equation 4 takes the value of 0.2, 0.3, 0.4 and 0.5 for every diameter of the structuring element tested. Percentage values after the reconstruction from masks generated through standard operators are shown in gray boxes, through viscous operators in black boxes.

Table 1. The table shows the mask images used in the diagrams in Figure 6

Index	Operation	S.E. Diameter	p
1	-	-	-
2	Standard Dilation	5	-
3	Standard Dilation	10	-
4	Standard Dilation	20	-
5	Standard Closing	5	-
6	Standard Closing	10	-
7	Standard Closing	20	-
8	Standard Closing	30	-
9	Viscous Closing	20	0.2
10	Viscous Closing	20	0.3
11	Viscous Closing	20	0.4
12	Viscous Closing	20	0.5
13	Viscous Closing	30	0.2
14	Viscous Closing	30	0.3
15	Viscous Closing	30	0.4
16	Viscous Closing	30	0.5
17	Viscous Closing	40	0.2
18	Viscous Closing	40	0.3
19	Viscous Closing	40	0.4
20	Viscous Closing	40	0.5
21	Viscous Closing	50	0.2
22	Viscous Closing	50	0.3
23	Viscous Closing	50	0.4
24	Viscous Closing	50	0.5

The ideal situation expected for a given mask would show a high percentage for the corresponding box in Fig. 6a and a low percentage for the one in Fig. 6b. This means that the object that should be considered linked to another is restored more than an other object in the same image not linked to anything else. The best results observed in the experiments are obtained with the connectivity defined by a viscous mask generated by a structuring element whose diameter length is 50 pixels and parameter p equals to 0.4 (at index 23) and by a structuring element whose diameter length is 40 and p equals to 0.3 and 0.4 (at indexes 18 and 19). For larger values of p, too much clustering appeared, so these results are not shown in the diagram. Previous experiments conducted on 8-bit images taken from the same Sloan survey and derived from the 16-bit images showed that $p = 0.5$, meaning that the diameter of the structuring element decreases as the square root of the image intensity, turned out to be the best value. This difference is probably due to some dynamic range compression in the conversion



(a)

(b)

Fig. 6. The diagrams show the percentage value of the reconstructed level of the adjacent component. At index 1, no mask image is used at all. Indices from 2 to 8 illustrate the case of masks generated through standard operators. Indices from 9 to 24 illustrate the case of mask images obtained through viscous closings. The settings of every mask are given in Table 1.

to 8-bit per pixel. Using the actual 16-bit images, features like the filaments appear to be fainter with respect to the main objects than in the 8-bit images. This means that more clustering is necessary in this case to preserve the fainter components, as reflected in p giving the best results for values lower than 0.5.

Regarding the behaviour of the masks generated by standard operators, they are obviously able to cluster merging galaxies at the same level as or higher than the viscous masks. The drawback is that they also cluster unwanted objects together. This is reflected in the diagram in the positions of indexes from 3 to 8. Note that at index 2 the percentages show a decent reconstruction of the galactic component while keeping other objects separate. However, it must be considered that a dilation with a structuring element with diameter 5, shown at index 3, also clusters the unwanted object. If masks generated from standard operators are used, the reconstruction looks very sensitive to the diameter: an increase of the length from 5 to 10 pixels modifies the output radically. In the viscous case, closer percentages are obtained when similar settings are used, like for example at indices 21, 22 and 23.

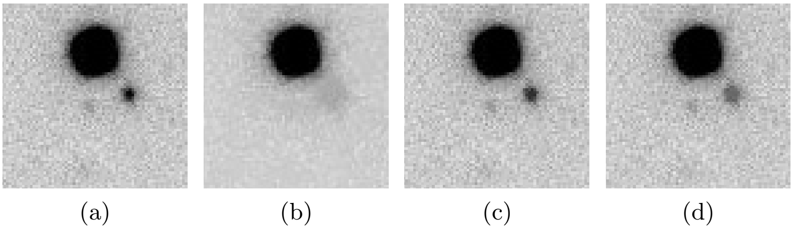


Fig. 7. (a) is a region of the original image; (b) shows the reconstruction when no mask is used; (c) and (d) show the reconstruction when a standard and a viscous closing were used as masks, respectively. The marker used for reconstruction is placed on the big star.

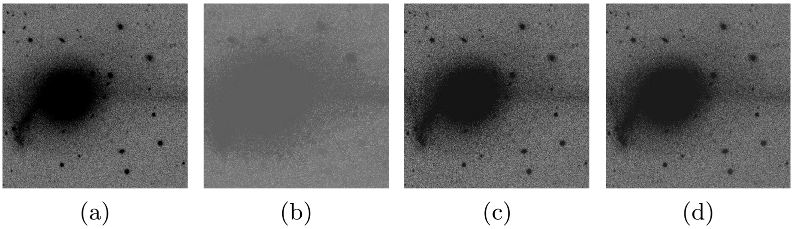


Fig. 8. (a) is a region of the original image; (b) shows the reconstruction when no mask is used; (c) and (d) show the reconstruction when a standard and viscous closing were used as masks, respectively. The marker used for reconstruction is placed on a galaxy on the right.

4.2 A Visual Evaluation

Figure 7 and Figure 8 represent two regions of Figure 1a, depicting a very critical situation composed of two stars very close each other and two merging galaxies. Image intensities have been clipped for the purpose of visualization. The markers were placed on the brighter star and on the rightmost galaxy. Two connectivity masks are compared here: the first is generated by a standard closing and the other by a viscous closing. Figure 7b shows that with no second-generation connectivity, the star component is weakly reconstructed: the small star is reconstructed up to the intensity value of the saddle point, or 0% in our graphs.

When a masks generated by a standard closing is applied, as in Figure 7c, the star component is reconstructed more than in the case of a mask generated by a viscous closing (73% vs. 38%). On the other hand, as in Figure 8, the galactic component reached thanks to a better preserved filament is reconstructed at the same level (∼80%) both for a mask generated by a standard closing and by a viscous closing. Generally speaking, for the viscous mask case, it is then observed that stellar components that are not interacting are clustered less, while at the same time the interacting galaxies are reconstructed to a level comparable to that obtained in the standard closing or even higher. This depends on the maximum size of the structuring element and on the value of p used.

5 Conclusions

In this paper, an evaluation of connectivity masks generated through viscous mathematical morphology operators has been presented. These masks work well in keeping nearby stars separate, while clustering together the faint filaments typically linking merging galaxies. Viscous operators clearly work better than the regular closing used in this experiment. The tuning of the parameters driving the size reduction will be subject of further studies. More experiments are currently being conducted to determine the optimal settings for both this parameter and the maximum size of the structuring element in a wider range of test cases. While more fine-tuning is needed, it is evident from this initial work that a better object separation is already obtained by viscous masks. Other clustering masks are also under investigation.

Mask-Edge connectivity [9] looks to be a promising tool to generate mask images that do not allow clustering in presence of sharp edges. K-flat filters [14] might be used as well to preserve faint components up to k intensity values more, but only when they show certain characteristic or shape features, such as elongation, for example, that can suggest that it is a filament. Moving away from mathematical morphology, first experiments have been started with PDE-based methods relying on Perona-Malik [15] diffusion, or variants that are driven more by image intensities than by just image gradients. These might be suitable to be used in astronomical imagery in some cases. Softwares performing object segmentation like SExtractor would benefit from the use of connectivity mask for a more reliable deblending of the objects of interest.

Acknowledgement. This work was funded by the Netherlands Organisation for Scientific Research (NWO) under project number 612.001.110.

References

1. Serra, J.: Connectivity on complete lattices. J. Math. Imag. Vis. 9(3), 231–251 (1998)
2. Braga-Neto, U., Goutsias, J.: Connectivity on complete lattices: new results. Comput. Vis. Image Underst. 85(1), 22–53 (2002)
3. 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)
4. Ouzounis, G.K., Wilkinson, M.H.F.: A parallel implementation of the dual-input max-tree algorithm for attribute filtering. In: Banon, G.J., Barrera, J., Braga-Neto, U.M., Hirata, N.S. (eds.) Proc. Int. Symp. Math. Morphology (ISMM 2007), São José dos Campos, Universidade de São Paulo (USP), Instituto Nacional de Pesquisas Espaciais (INPE), vol. 1, pp. 449–460 (October 2007)
5. Bertin, E., Arnouts, B.: Sextractor: software for source extraction. Astronomy and Astrophysics 117(suppl. ser.), 393–404 (1996)
6. Vachier, C., Meyer, F.: The viscous watershed transform. J. Math. Imag. Vis. 22(2-3), 251–267 (2005)
7. Vachier, C., Meyer, F.: News from viscousland. In: Banon, G.J., Barrera, J., Braga-Neto, U.M., Hirata, N.S. (eds.) Proc. Int. Symp. Math. Morphology (ISMM 2007), São José dos Campos, Universidade de São Paulo (USP), Instituto Nacional de Pesquisas Espaciais (INPE), vol. 1, pp. 189–200 (October 2007)
8. Maragos, P., Vachier, C.: A pde formulation for viscous morphological operators with extensions to intensity-adaptive operators. In: Proc. Int. Conf. Image Proc. 2008, pp. 2200–2203. IEEE (2008)
9. Oosterbroek, J., Wilkinson, M.H.F.: Mask-edge connectivity: Theory, computation, and application to historical document analysis. In: ICPR 2012 (2012)
10. Vincent, L.: Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. IEEE Transactions on Image Processing 2, 176–201 (1993)
11. Abazajian, K.N., Adelman-McCarthy, J.K., Agüeros, M.A., Allam, S.S., Allende Prieto, C., An, D., Anderson, K.S.J., Anderson, S.F., Annis, J., Bahcall, N.A., et al.: The Seventh Data Release of the Sloan Digital Sky Survey. The Astrophysical Journal Supplement Series 182, 543–558 (2009)
12. 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)
13. Wilkinson, M.H.F.: A fast component-tree algorithm for high dynamic-range images and second generation connectivity. In: Proc. Int. Conf. Image Proc. 2011, pp. 1041–1044 (2011)
14. Ouzounis, G.K., Wilkinson, M.H.: Hyperconnected attribute filters based on k-flat zones. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(2), 224–239 (2011)
15. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)

Discrete Simulation of a Chladni Experiment

Frédéric Rieux[1,2]

[1] LIRMM, Université Montpellier 2, 161 rue Ada,
F-34392 MONTPELLIER, France
[2] I3M, Université de Montpellier 2 c.c. 51
F-34095 Montpellier Cedex 5, France

Abstract. Digital diffusion processes have been introduced to capture information about the neighborhood of points in a digital object. The properties of these processes give information about curvature, about specific symmetries and particular points on the discrete set. The evolution of diffusion is governed by the LAPLACE-BELTRAMI operator which presides to the diffusion on the manifold, as for example random walks. In this paper, we will study the discrete Laplacian operator defined on pixels in order to understand the symmetries and extract their intersections. This will lead to the identifications of particular points or information about geometry of a digital set.

1 Introduction

Can you hear the shape of a drum?

Spectral geometry is a domain that studies the link between spectral decomposition of Laplace-Beltrami operator on a Riemanian surface and its geometric characteristics. This domain is then related to the question: if we know the spectral decomposition of an object like a drum, can we deduce its geometrical structure? The study of eigenvectors and Laplace-Beltrami operator is a common problem in mathematics and there exists a large amount of works in the XVIIIth and XIXth centuries that handle the subject for Riemanian surfaces. Although this approach is particularly efficient, there exists few works in digital processing that use spectral decomposition to study discrete shapes.

Moreover, in geometry processing applications, the spectral decomposition on mesh structures has been used recently, for example lately by SUN, OVSJANIKOV and GUIBAS [11] and GEBAL, BÆRENTZEN, AANÆS and LARSEN [7]. A generalization of those approaches has been proposed recently in [4].

Bruno Levy in [10] has proposed to use the classical notion of spectral geometry to study the geometry of meshes structures. This approach is based on the discretisation of the Laplace-Beltrami operator, and he studied the zero-values of the greatest eigenvectors to characterize the meshes.

Those approaches are particularly interesting and offer us an important tool widely approved in mathematics and we propose in this paper to recover those classical notions for digital shape. However, on mesh structures the classical approach is based on discretisation of Laplace-Beltrami operator. We will use a

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 496–507, 2013.

discrete diffusion process to define directly a discrete Laplace operator and then we will study the eigenvectors decomposition of this operator. The Laplace-Beltrami operator governs different diffusion processes, unrelated at first sight, heat flows, acoustic and small mechanical deformations.

We introduced in [6] an auto-adaptive digital process which captures information about the neighborhood of a point in a shape. This process is discrete and take into account the geometry of the shape. We set walkers up on a digital object, and we observe the possibilities for them to walk along the discrete set. This approach allows us to compute weights for each point, according to the time spent at each point. In [6], we let walkers diffuse on a digital function to recover information about the curvature, enabling them to define derivative and curvature estimator robust to noise. The geometrical information yield by the process allows us to extend the diffusion to higher dimension sets, in particular on digital stars.

In this paper, we will investigate the properties of the auto-adaptive process on subset of $\mathbb{Z}^2$. We will set up walkers on a digital object and we will study the eigenvectors of the discrete Laplacian to describe and understand its geometry. The aim is to understand the geometry of discrete shapes given by pixels, using a discrete digital process.

A possible application is the study of digital stars. Basic properties of digital stars have recently been given by Valentin E.Brimkov and Reneta P. Barneva [3]. In this paper, one of the questions raised is to find a “center“ for a digital star. We study the eigenvectors of the discrete Laplacian defined in Section 2.2 in order to find a subset included in the star that contains good candidates for the solution. We define *Centroids* as an intersection of many self directions (Section 3.1). We also discuss the convergence of this topological approach, recovering Chladni acoustic plates on a discrete set.

2 Diffusion Processes

2.1 Heat Diffusion

The heat kernel k_t of a manifold M maps a pair of points $(x, y) \in M \times M$ to a positive real number $k_t(x, y)$, which describes the transfer of heat from y to x in time t. Starting from a (real) temperature distribution T on M, the temperature after a time t at a point x is given by a convolution of the initial distribution with the diffusion kernel:

$$H^t\, f(x) = \int_M f(y)\, k_t(x, y)\, dy.$$

The heat equation is driven by the diffusion process, while the evolution of the temperature in time is governed by the (spatial) Laplace-Beltrami operator Δ_M: $\frac{\partial f(t,x)}{\partial t} = -\Delta_M f(t, x)$ which presides to the diffusion on the manifold, just as random walks.

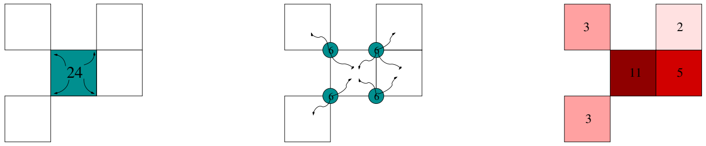
The heat equation, lately used by Sun, Ovsjanikov and Guibas [11], yields information about the geometry of the manifold. We have proposed in a last paper a digital diffusion process [6] which is adaptive to the geometry of a digital object. Using this digital process, we aim to recover particular directions of diffusion in a digital object, like a digital star, to find symmetrical points of diffusion, which we will call the centroid of the digital object.

2.2 Auto-adaptive Process

We set up the diffusion of a walker wandering on the discrete object, in order to recover its local geometry. This auto-adaptive process is a local Markov process, a stochastic process in discrete time where the probability of a transition between pixels depends only on the immediate past. We propose to define a diffusion process using the topological adjacence of pointels in a discrete set.

Definition 1 (Standard adaptive process). *[6]*
We denote standard adaptive process*, the* Markov *chain with the probabilities of transition from pixel to pixel defined by the composition of transitions from a pixel of the set to its four pointels and then from a pointel to adjacent pixels of the set constrained by:*

1. Equiprobability $\frac{1}{4}$ *to go from a pixel to one of its pointels;*
2. Equiprobability *to move from a pointel to an adjacent pixel of the set.*



(a) From pixel to pointels.

(b) From pointels to adjacent pixels.

(c) On **Z**, the binomial coefficients are recovered.

Fig. 1. A walker has a probability of $\frac{1}{4}$ to pick up a corner of a pixel, then $\frac{1}{k}$ to select a pixel among k possible

This standard choice of walkers with no memory and only local knowledge is the same as the famous short-sighted drunken man*. We denote this process A_s^m for a walker starting at any given point, with m being the number of iterations in the process.*

The 1-step Markov process transition matrix A_s is simply a weighted version of the adjacency matrix of the digital object $\mathcal{M}$ (Fig.1). We denote $u_0 = u(m = 0,\ x)$ the number of walkers on the digital object at time 0 in x. We denote X_m the Markov chain defined by Def. 1 of length m. Let h be the step of digitalization: it is the length of the grid of the digital object. The diffusion time in the Markov chain in terms of h after m steps is $t = \frac{m\,h}{c}$ with $m \in \mathbb{N}^*$ and c a characteristic “speed” of the process.

Let $u(m,\ x)$ be the number of walkers at x after m steps. We can therefore write:

$$u(m,\ x) \ = \ \mathbb{E}_x u_0(X_m)$$

with $\mathbb{E}_x u_0(X_m)$ being the expected value to be in x with $u_0(X_m)$ walkers after m steps. As in the case of the heat diffusion, we have the following diffusion equation:

$$\frac{\partial u(m,\ x)}{\partial m} \ = \ \frac{1}{h}\left(A_s \ - \ Id\right) u(m,\ x)$$

We define the discrete Laplacian $\Delta_M \ := -\frac{1}{h}\left(A_s \ - \ Id\right)$

3 Spectral Decomposition

Ernst Chladni was a physician who studied vibrations of thin metal plates. Spreading sand over it, he observed interesting patterns corresponding to the accumulation of sand. In [10], Bruno Levy used the discretization of the Laplace operator on digital meshes sets, in order to recover the contour of the eigenvectors and therefore get similar patterns to the Chladni experiments. The study of the eigenvectors has been recently used in [11,7,4]. The aim of this paper is to study a particular digital diffusion process defined in [6] to compute without discretization the contour of the eigenvectors on digital sets. In the next section 3.1, we will propose a simple application to recover, on digital stars with particular symmetry, the centers of the stars using the contour of discrete eigenfunctions. Then we will discuss the convergence of this approach by recovering on digital plates the acoustics waves observed by the physician Ernst Chladni.

3.1 Symmetry Detection Using Diffusion Direction

The auto-adaptive process defined in Sec. 2.2, is adaptive to the geometry. Indeed, a walker gives local information about the possibilities to explore the set. We have therefore a characterization for the neighborhood of a pixel on the set. Let $\lambda_0,\ \lambda_1,\ldots,\ \lambda_n$ be the eigenvalues of A_s the transition matrix of the auto-adaptive process sorted in decreasing order. According to the definition of a stochastic process, the largest eigenvalue is $\lambda_0 \ = \ 1$ (see Theorem. 2). We denote $sp(A_s)$ the spectral space of the matrix A_s, and we note ϕ_i the i-th eigenvector.

Theorem 2. *Perron-Frobenius[2]*
Let P be a square matrix positive and irreducible. Then there exists an eigenvalue r with the following properties:

- $r \ > \ 0$
- *for each eigenvalue $\lambda \in sp(P)$, $|\lambda| \leq r$*
- *The eigenspace associated to r has dimension one*

We studied the eigenvector decompositions of particular sets to recover geometrical information (see Figure 2). As we can see in this example, there are floors corresponding to eigenvalues close to each other. The number of these eigenvalues correspond to the numbers of branches in the digital star related to the position of the first sharp gap in the eigenvalues.

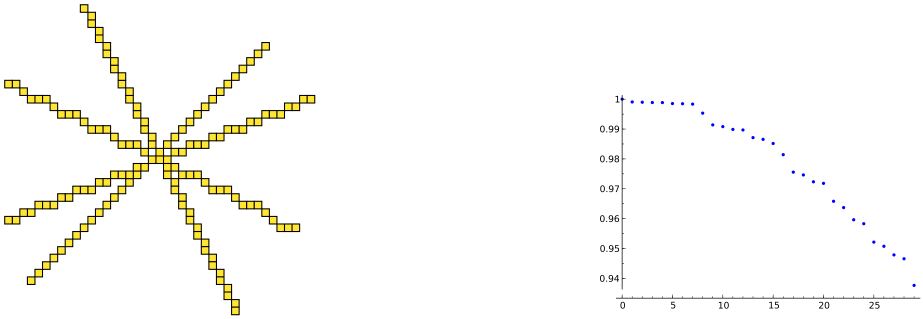


Fig. 2. Computation of the first eigenvalues of a digital star. The eigenvalues are sorted in decreasing order. We can see a gap between the seventh and eighth values corresponding to the number of branchs in the digital star. On right side the y-axis correspond to the value of eigenvalue and x-axis correspond to the index of the eigenvalue in decreasing order.

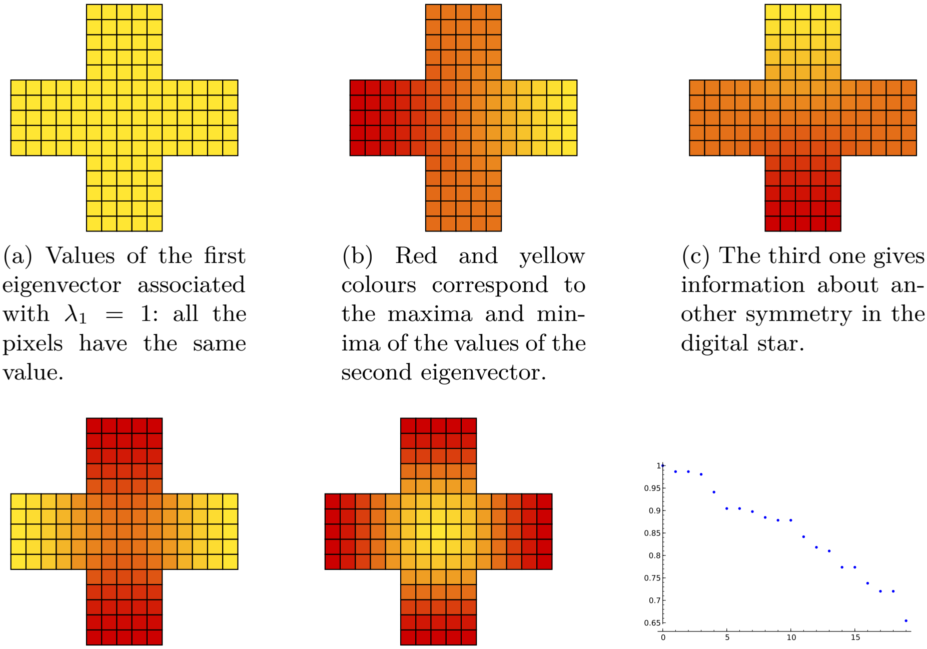
As in the case of the heat kernel, we have a spectral decomposition of the kernel of the auto-adaptive process k_m:

$$k_m = \sum_{y \in M} e^{-m\lambda_i} \phi_i(x) \phi_i(y)$$

The process is characterized by the values of the eigenvectors associated to each point of the digital set. For the first eigenvalues we sketch the values of the eigenvectors (see Figure 3a, Figure 3b, Figure 3c, Figure 3d). The eigenvectors give us geometrical information about particular diffusion directions in the set.

As we can see in Figure 3e, the values of the eigenvector just after the gap seem to give information about the center of the star. The eigenvectors associated with the larger eigenvalues characterize the symmetry in the digital set. The gap characterizes a particular geometrical symmetry: with the help of physical analogies, we can identify each symmetry as the direction of vibrations in the object. The gap occur when all the branches vibrate at the same times. We segment the object into constant sign eigenvector values and we compute their intersection to extract particular geometrical points in a discrete shape. We then fuse the information shared by each eigenvector.

Definition 3 (Vector Intersection).
Let A_s be the matrix of the auto-adaptive process, let P be the matrix of eigenvectors and D the diagonal of their eigenvalues, sorted in decreasing order, such



(a) Values of the first eigenvector associated with $\lambda_1 = 1$: all the pixels have the same value.

(b) Red and yellow colours correspond to the maxima and minima of the values of the second eigenvector.

(c) The third one gives information about another symmetry in the digital star.

(d) The fourth one: when the values of the opposite petals are maximal the others are minimal.

(e) The eigenvector just after the gap: the center of the star appears clearly.

(f) First eigenvalues of a digital Swiss cross sorted in decreasing order.

Fig. 3. Values of the first five eigenvectors of the auto-adaptive process. The values of these eigenvectors change from yellow (minimum) to red (maximum). The digital symmetries in the Swiss cross appear clearly

that $A_s = PDP^T$. *That is to say that* $P = (\phi_0|\phi_1|\dots|\phi_n)$ *with* ϕ_i *the i-th eigenvector. We denote the* intersection *of the* ℓ *first vectors in* x, $I_x^{(\ell)}$ *being the following values:*

$$I_x^{(\ell)} = \sum_{i=0}^{\ell} \phi_i^2(x)$$

We compute for each point of the digital set, the value $I_x^{(\ell)}$ starting with $\ell = 1$, and we record the position x of the minimal value at each iteration. For the first iterations, the position of the minimal value of $I_x^{(l)}$ will move until the intersections stabilize around the same point. We simply record the position of this possible center and stop the iteration when the min stops moving.

Definition 4 (Centroid position).
Let $\mathcal{I}^{\ell} = \{x \in \mathcal{D}/I_x^{(\ell)} = \min_{y\in\mathcal{D}}(I_y^{(\ell)})\}$ *be the set of positions where the minimal value of* $I_x^{(\ell)}$ *is reached for* $x \in \mathcal{D}$*. We denote a* centroid *the set of points such that* $\mathcal{I}^{\ell} = \mathcal{I}^{\ell+1}$*.*

Remark 5. We introduce in the definition of centroids a specific geometrical point given by the intersection of self directions. This is linked to the question asked by Mark Kac in 1966,“ Can you hear the shape of a drum? “ The eigenvectors could be seen as waves striding the discrete shapes and recording geometrical properties. Moreover, in 1992 [8] C. Gordon, D. Webb et S. Wolpert proposed isospectral decomposition of different planar domains. Therefore, the information given by centroids is not generally clear. However, it is an interesting benchmark to study or to research particular structures such as the simple and symmetrical objects like the "digital stars".

4 Discussion about Convergence

4.1 Digital Stars

The process we used to compute the spectral domains of a shape is based on the diffusion of a walker using the topology of a discrete shape, contrarily to the study of the heat diffusion in a shape. A large proportion of the studies taking place in mesh structures used a continuous process and gave an approximation of the Laplace operator on the meshes. Therefore the convergence of the methodology is related to the efficiency of the discretization. Our approach cuts short the problem of digitalization but we still need to study behavior when the grid of digitalization of the discrete shape converges toward zero. In a generic case, we are not able to characterize the spectral decomposition of a shape (see Remark. 5). But if the object has particular symmetry we can prove that the centroid is a geometrical particular point.

Definition 6 (Simple Star).
We denote a simple star, the gaussian discretisation of a star with two discrete lines which are symmetries axes.

Proposition 7. *Let* $\mathcal{D}$ *be a digital simple star, if* $\mathcal{I}^{\ell} = \mathcal{I}^{\ell+1}$*, the centroids* $\mathcal{I}^{\ell}$ *is a center of the digital star.*

Remark 8. The existence of the eigenvalue which cuts a simple star on the geometrical center can be seen with physical analogies. Let $\mathcal{D}$ be a digital star, seen as the membrane of a drum. We hit the star at several positions and we listen to the vibration of the sound in the star. Each eigenvector of the diffusion process gives a particular vibration mode of the branches of the star.

Proof. Let λ_{k+1} be the eigenvalue such that the minimum of the eigenvector ϕ_{k+1} associated with λ_{k+1} is reached at the center of symmetry p_{k+1} of the digital star. The eigenvector ϕ_{k+1} is such that all the branches vibrate at the same time (we can see examples in Section 3.1).

Let λ_ℓ be an eigenvalue such that $1 > \lambda_\ell > \lambda_k$. Then ϕ_ℓ gives a symmetry of the digital star, e.g the positive and negative values of the eigenvector cut the star in several parts so that the zero values are on the symmetry axis. We send the minimal values of the eigenvector on the symmetry axes in computing $N(\phi_\ell) = \phi_\ell^2(x)$ for $x \in \mathcal{D}$. By the property of the digital center, two branches in vibration modes have a common point p_ℓ the hit point of the sound diffusion. On the supposition that $\mathcal{D}$ is a digital simple star, the symmetry axes are digital straight segments included in the star. Therefore all the points of the branches in vibrations can be joined by a straight segment to $p_\ell \in \mathcal{I}^\ell$.

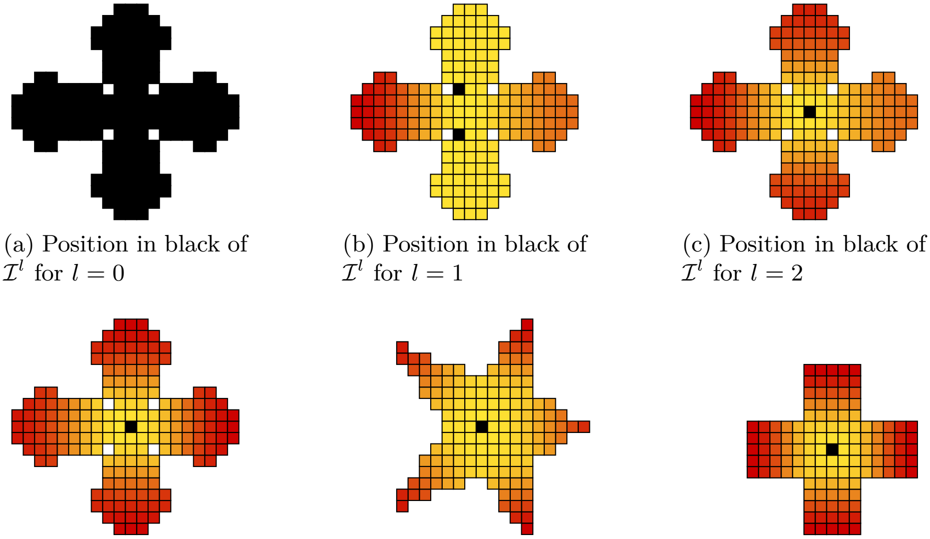
Let $I_x^{(\ell)} = \sum_{i=0}^{\ell} \phi_i^2(x)$ be the intersection of the eigenvectors for $i \in [1..\ell]$. Let $\mathcal{I}^\ell$ be the position of the minimum values after ℓ intersections. The minimal values of the norm of each eigenvector, by construction, are on the symmetry axis. Then the position of the minimal value after ℓ intersections, is a point close to each symmetry axis. We suppose now that $\mathcal{I}^\ell = \mathcal{I}^{\ell+1}$. Let $p_{\ell+1} \in \mathcal{I}^{\ell+1}$ be a position of the minimal value. Then $p_{\ell+1}$ are close to all the symmetries axes, it sees the points of the branches in vibration for the self direction $\phi_{\ell+1}$ and it belongs to another self direction. Suppose now that a branch exists in the star, not seen by $p_{\ell+1}$. Then, if we call $C_\mathcal{D}$ the set of solutions for the center of the star, $C_\mathcal{D} \subsetneq \mathcal{I}^{\ell+1}$ and $p_{\ell+1} \notin C_\mathcal{D}$. Then there exists a symmetry in the star at index e such that $1 > \ell > e > k+1$ and cut $\mathcal{I}^{\ell+1} = \mathcal{I}^e \cup R$ with R the rest of the minimal value: $p_{\ell+1} \in R$. We have also $\mathcal{I}^\ell = \mathcal{I}^e \cup R$, then the self directions $\phi_{\ell+1}$ cut $\mathcal{I}^\ell$ exactly as ϕ_ℓ for $\mathcal{I}^{\ell-1}$ then $\phi_\ell = \phi_{\ell+1}$, contradiction. □

Examples of these results is given Figure. 4.

4.2 Chladni Experiments

The experiment of Chladni [5] consists in spreading sand on a metal plate laid on its center on top of a vibrating jar. The frequency of the plates vibrations induces accumulation of sand on the plate. Ernst Chladni remarks that for different frequency modes sand forms complex patterns (Figure. 5). This phenomenon is explained by the nodes of the wave (the non-moving parts) which form a complex pattern. The wave is also modeled by an eigenvector of the Laplacian. The nodes of a wave are the null-values of the associated eigenvector. To discuss the convergence, we propose a simulation of the Chladni experiment using the discrete process. The aim is to recover the same patterns in order to give an idea of the convergence of the discrete process on a shape. The idea is to use a discrete plate with an appropriate length of discretization to see the convergence of the Chladni plate eigenvectors contours.

We propose in Figure. 6, the results of the pattern found on a plate with a step of digitalisation of $\frac{1}{100}$. We recover all the complex patterns observed by Chladni. These results are already known for discrete shapes and the contour can be easily computed with Bessel functions. Moreover it is interesting to see that our Laplace operator recovers the exact patterns. To convince about the efficiency



(a) Position in black of $\mathcal{I}^l$ for $l = 0$

(b) Position in black of $\mathcal{I}^l$ for $l = 1$

(c) Position in black of $\mathcal{I}^l$ for $l = 2$

(d) Position in black of $\mathcal{I}^l$ for $l = 3$: $\mathcal{I}^l = \mathcal{I}^{l+1}$

(e) Position in black of $\mathcal{I}^l$ for $l = 5$.

(f) Position in black of $\mathcal{I}^l$ for $l = 3$

Fig. 4. Examples of stars centroids detection, with l being the number of eigenvectors used to find the center

of our approach, we compute the contours of eigenvectors on the discretisation of a triangle and a guitar with $h = 1/100$. These are interesting shapes because, contrarily to the square plates, Bessel functions could not compute the contour (see Figure. 8).

5 Conclusion

In this paper, we studied the spectral decomposition of a digital process based on topological adjacencies between pixels. This approach is related to a large number of works which all studied the heat diffusion on mesh structures in order to extract geometrical characterization. Discretization is required to study the eigenfunctions of the meshes. We introduce a digital diffusion process on voxels based on topological adjacencies between voxels to recover information about the geometry of the shape. On very simple objects like digital stars, we segment the object into constant sign eigenvector values and compute their intersection to recover the center called centroids. We discuss the convergence of the process when the digitalization step converges towards zero. Considering the convergence of the discrete Laplacian, this approach recovers the exact patterns of the Chladni experiment.

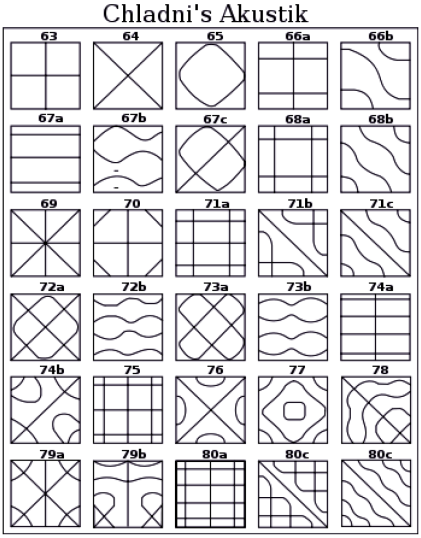
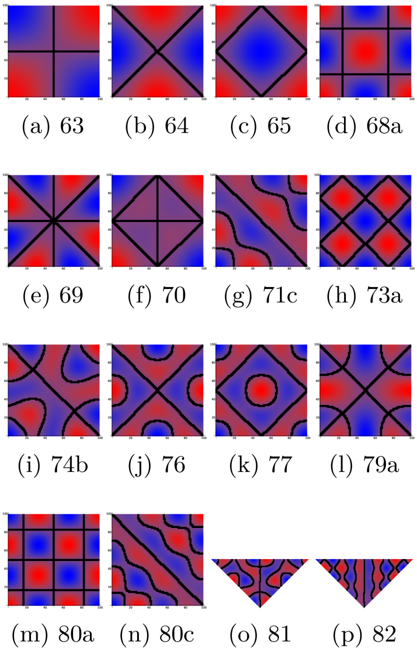


Fig. 5. Accoustic waves pattern of the Chladni experiments on a metal square plate with $h = 1/100$ (100x100 pixels) [5].



(a) 63 (b) 64 (c) 65 (d) 68a

(e) 69 (f) 70 (g) 71c (h) 73a

(i) 74b (j) 76 (k) 77 (l) 79a

(m) 80a (n) 80c (o) 81 (p) 82

Fig. 6. Discrete patterns computed on a discrete plates of digitalisation step $\frac{1}{100}$.

Fig. 7. Comparison between the discrete patterns and the simulated pattern. We draw the eigenvector values (red to blue) and the zero values in black. We recover the complex pattern of the continuous waves on a metal plate. Considering the convergence of the discrete Laplacian, this approach recovers the exact patterns of the Chladni experiment. Due to lack of space, we propose only few examples of patterns computed.

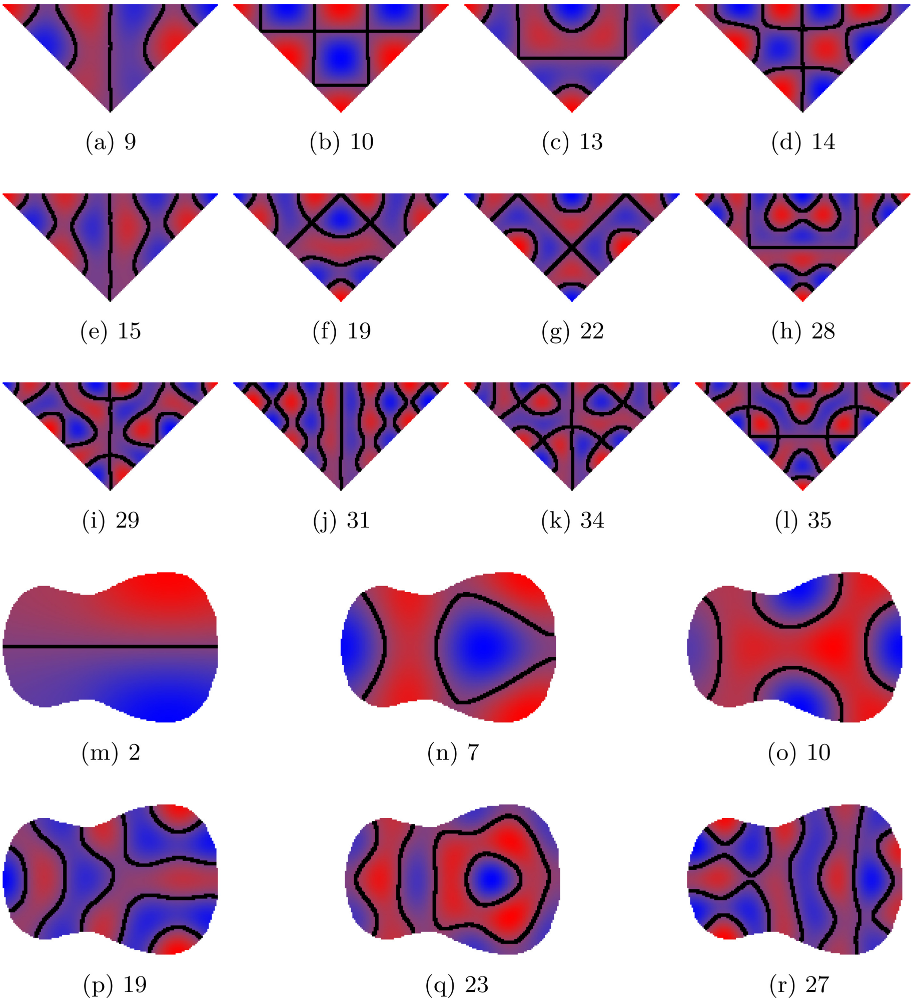


Fig. 8. We recover the complex pattern of the continuous waves on a triangle and on a guitar. Considering the convergence of the discrete Laplacian, this approach recovers the exact patterns of the Chladni experiment.

In future works, we will use this diffusion process to compute the properties of digital curves. We will show that this geometrical estimators used as a convolution mask in order to ponderate locally the shape is multigrid convergent, meaning that the geometrical estimator converges toward the geometrical values of the continuous shape inherent to digitalization.

Acknowledgement. We would like to thank Pascal Azérad[1] for his help on Markov processes, Christian Mercat[2] and Christophe Fiorio[3] for the useful and accurate comments provided.

References

1. Barneva, R.P., Brimkov, V.E., Hauptman, H.A., Natal Jorge, R.M., Tavares, J.M.R.S. (eds.): CompIMAGE 2010. LNCS, vol. 6026. Springer, Heidelberg (2010)
2. Billingsley, P.: Convergence of probability measures, 2nd edn. Wiley Series in Probability and Statistics: Probability and Statistics. John Wiley & Sons Inc., A Wiley-Interscience Publication, New York (1999)
3. Brimkov, V.E., Barneva, R.P.: Digital stars and visibility of digital objects. In: Barneva, et al. (eds.) [1], pp. 11–23
4. Bronstein, M.M., Bronstein, A.M.: Shape recognition with spectral distances. IEEE Trans. Pattern Anal. Mach. Intell. 33(5), 1065–1071 (2011)
5. Chladni, E.F.F.: Traité d'acoustique. Chez Courcier (1809)
6. Fiorio, C., Mercat, C., Rieux, F.: Curvature estimation for discrete curves based on auto-adaptive masks of convolution. In: Barneva, et al. (eds.) [1], pp. 47–59
7. Gebal, K., Bærentzen, J.A., Aanæs, H., Larsen, R.: Shape Analysis Using the Auto Diffusion Function. In: Konrad, et al. (eds.) [9], pp. 1405–1413
8. Gordon, C., Webb, D.L., Wolpert, S.: One cannot hear the shape of a drum. Bull. Amer. Math. Soc. (N.S.) 27(1), 134–138 (1992)
9. Konrad, P., Marc, A., Michael, K. (eds.): Symposium on Graphics Processing. Eurographics Association (2009)
10. Lévy, B.: Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry. In: SMI, p. 13. IEEE Computer Society (2006)
11. Sun, J., Ovsjanikov, M., Guibas, L.: A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. In: Konrad, et al. (eds.) [9], pp. 1383–1392

[1] University Montpellier 2

[2] University Lyon 1

[3] University Montpellier 2

Automated Quality Inspection of Microfluidic Chips Using Morphologic Techniques

Thomas Schwarzbauer[1,2], Martin Welk[2], Chris Mayrhofer[1],
and Rainer Schubert[2]

[1] Sony DADC Austria, Sonystrasse 20, 5081 Anif/Salzburg, Austria
{thomas.schwarzbauer,chris.mayrhofer}@sonydadc.com
[2] University for Health Sciences, Medical Informatics and Technology (UMIT),
Eduard-Wallnöfer-Zentrum 1, 6060 Hall/Tyrol, Austria
{thomas.schwarzbauer,martin.welk,rainer.schubert}@umit.at

Abstract. We apply morphological image processing for quality inspection of microfluidic chips. Based on a comparison of measured topographies with design data, we provide a coherent solution to four central tasks in the quality assessment of injection moulded polymer devices: determination of channel depth, identification of burrs, calculation of transcription accuracy, and detection of defective regions. Experimental comparison to manual quality inspection procedures demonstrates the good performance of the proposed automated method, and reveals its clear advantages in terms of objectivity and reliability.

1 Introduction

Visual quality inspection of industrial products has been an important application field for image processing from its beginnings, see e.g. [15,24]. Manual quality inspection by humans faces numerous problems [14]. For example, human experts require training and their skills take a lot of time to develop. Even between well-trained individuals, results tend to be observer-dependent. Furthermore, the inspection task can be tedious and difficult. As [15] points out, this conventional kind of quality control is not only slow and costly, but also leads to high scrap rates and does not assure high quality.

In many mass production manufacturing facilities the actual goal is a 100 % quality assurance [15] which is often unfeasible in a manual inspection setting. Hence, there is an ever-increasing demand for automated visual quality inspection.

In this paper, we are concerned with the development of methods for semi-automated quality inspection of microfluidic chips. Being used e.g. for in-vitro diagnostics, life sciences research and medical technology, these chips constitute a rapidly growing market. A microfluidic chip is a polymer plate incorporating channels and filters through which liquid or emulsions can propagate and are led to well-defined compartments where reactions can take place. Channel cross-sections typically measure some tens of micrometers, while filter structures go down to few micrometers in size. For an example, see Fig. 1.

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 508–519, 2013.

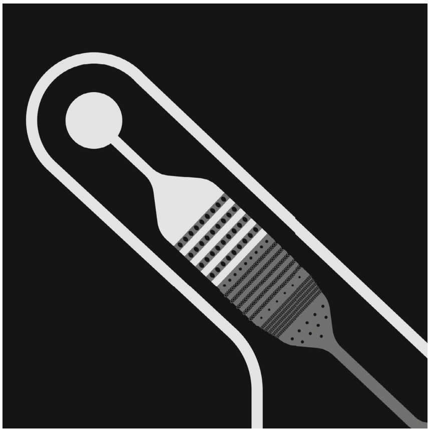


Fig. 1. Detail from a microfluidic chip (design data) featuring channel and filter structures. Grey-values represent depth values on a scale that spreads about 60 µm.

A typical production process involves injection moulding of a half-product plate with open channels and filter structures embossed in its surface, which is afterwards covered by an unstructured even plate. The inspection of the half-product surface is in the focus of our work.

Quality requirements for these products are extremely high due to the fact that they are often used for the processing of unique and sensitive probes. It is of utmost importance to ensure that no probes are wasted due to malfunctions of the chip, and that results of analysis have a high degree of reliability. As a consequence, a thorough quality inspection – ideally, a 100 % quality check – is crucial. At the same time, in order to keep scrap rates low, chips should be discarded only if they feature defects which indeed impair the function of the appliance.

Reviewing quality inspection of surfaces across industries, [24] classifies inspection methods with regard to the underlying features: statistical, structural, filter-based and model-based.

Statistical approaches [24] utilize the spatial distribution of intensity values in images. A variety of statistical features are available and applied in literature for visual inspection. These techniques comprise approaches like histogram statistics, e.g. Ng [16] using global thresholding by Otsu [18], co-occurrence matrices, e.g. Asha et al. [1], Novak and Hocenski [17], as well as registration-based techniques, e.g. Chiou and Zhang [2], Tait et al. [20], Ibrahim et al. [8] and Leta et al. [9].

Structural approaches [24] use texture primitives and their spatial arrangement to analyze images [22]. Apart from primitive measurement, edge features and skeleton representation, also morphological operations are widely used for quality inspection purposes [24]. Elbehiery et al. [4] applied morphological techniques for ceramic tile inspection. Different defects, like chips, cracks or scratches can be detected by applying the proposed approaches, which are all sequences of different image processing techniques, like edge detection, noise reduction and morphology operations. To improve the capability of a morphological approach, Zheng et al. [26] tried to identify defects on aluminum surfaces by applying genetic algorithms in order to estimate the optimal morphological parameters, a segmentation threshold and a noise reduction threshold. Structural approaches based on morphology are also applied in steel quality inspection by Wiltschi et al. [23].

Filter-based techniques [24] are characterized by the application of filter banks and the computation of the energy of the filter responses. Methods in this category can be divided into techniques applied in spatial domain, frequency domain and joint spatial-frequency domain. Gabor filters were applied by Tsai et al. [21] to detect defects on tile, wood and fabric surfaces. A similar approach was proposed by Lin et al. [11]. They identified defects on optical lenses.

Model-based approaches [24] comprise fractal models [12], random field models [10] or texem models [25].

Our Contribution. We describe a system for the inspection of the surface quality of microfluidic chips. It is based on the comparison of topographic data obtained by confocal microscopy to a reference dataset derived from the design model. With a registration step as preprocessing, morphological techniques in combination with segmentation steps provide the core functionality of defect detection, which is in the focus of the present paper. Morphological operations serve as an efficient tool e.g. for noise reduction, recognition of structures and elimination of biasing areas.

Relying on well-established image processing techniques, the contribution of this work is to combine these methodologies into a coherent concept that goes all the way down from the specification of the particular quality inspection task to a practically usable solution.

2 Prerequisites

2.1 Data Acquisition and Problem Statement

In order to assess the conformity of the chip under inspection with its design, the chip is imaged by a confocal microscope at a resolution around 1 µm. This resolution is necessary in order to resolve the finest filter structures on the chip. From the measured data, a topographic map of the chip surface is created. This depth map can be treated as a 2D greyscale image. By design, channel and filter structures on the chip occur in a few discrete depth steps – besides the overall surface level, typically two to three different depth levels are involved.

Given the high resolution, measurement of the entire chip surface is precluded by both storage demand and measurement time. For this reason, the process is applied to regions of interest where critical channel and filter structures are located.

The processing is done by a combination of statistical and structural methods. In order to enable comparison between measured data and the design dataset, both images will have to be aligned. This task is known in image processing as image registration.

Using mainly morphological techniques, the depth map data is then evaluated with regard to several predefined quality criteria. Characteristics like channel depth, accuracy of transcription, and burr are of central interest here. Accuracy of transcription is meant to describe particularly how well filter structures are moulded. Filters typically consist of arrays of pillars, and it is important that these columns reach the overall surface level in order to get in touch with the closing plate. So the main criterion of transcription is the height of the columns. Burrs are raised regions adjacent to channel edges caused by material accumulations after injection moulding. A special challenge here is to discriminate between real burrs and artifacts of the confocal microscope scanning procedure. Transcription accuracy and burrs are indicators for possible functional problems of the chip.

Morphological techniques lend themselves as an excellent tool for the evaluation step because they do not only provide direct solutions to most of the shape processing tasks that occur herein, but also because of their simplicity and speed, which is crucial in view of the large amounts of data that need to be processed. Due to the layered design of the chips, set-valued morphological operations are suitable for our purposes.

2.2 Image Registration

Image registration denotes the task to establish a spatial transformation between the domains of two images by which corresponding points in both images are taken to the same locations. [5] For this to make sense it is understood that both images represent the same scene and differ by time of capture, viewpoint, and/or image modality [27]. Registration problems occur in multiple variations differing e.g. by dimensionality, registration basis, the realm of admissible transformations; registration algorithms differ further in their optimization procedures, and degree of interactivity [13].

In terms of these criteria, the registration problem we face can be described as 2D/2D because the depth dimension in both datasets is treated as (grey-) value. The transformation can be assumed as rigid because the main source of misalignment is the positioning of the probe under the confocal microscope which may be shifted, rotated and tilted, while it is not expected that deformations like shearing or bending of the chip itself as compared to its design model will be large enough to be observable as nonrigid within a region of interest as handled in the registration step. Furthermore, our registration problem is global (with regard to the region of interest), and automatic.

Image registration is nowadays a well-studied problem, and algorithms for a large variety of settings are available.

2.3 Morphology

Once the datasets are aligned, our further processing uses the standard operations of set-valued morphology, see e.g. [6], namely erosion $\ominus$, dilation $\oplus$, opening $\circ$ and morphological gradient ϱ_B with a structure element B as given by

$$A \ominus B = \{z \in E \mid B_z \subseteq A\} , \tag{1}$$

$$A \oplus B = \{z \in E \mid B_z \cap A \neq 0\} , \tag{2}$$

$$A \circ B = (A \ominus B) \oplus B , \tag{3}$$

$$\varrho_B(A) = (A \oplus B) \setminus (A \ominus B) , \tag{4}$$

where B_z denotes the translate of B to the location z.

2.4 Otsu Segmentation

Otsu segmentation [18] is a threshold-based segmentation method with automatic adaptation of thresholds to the image histogram. In the simplest case of a two-class (e.g. foreground-background) segmentation, a single threshold is chosen such that the ratio $\sigma^2_{\mathrm{inter}}/\sigma^2_{\mathrm{within}}$ of the variance $\sigma^2_{\mathrm{inter}}$ between classes and the variance $\sigma^2_{\mathrm{within}}$ within classes becomes maximal. Given the histogram, the threshold can be computed by an exhaustive search. The method is easily extended to more than two classes by selecting two or more thresholds; however, the cost of exhaustive search increases exponentially with the number of thresholds.

3 Methodology

In the registration step, the design depth-map is kept fixed, and the measured dataset is subject to a rigid transformation. This distribution of roles ensures that structures to be analysed later on are always found at the same positions. The transformation is determined such as to minimize the L_2-distance [3] of the gradient magnitudes between the fixed and transformed images. To this end, we apply a rigid multiresolution approach, see e.g. [5]. It is important here to work with gradients since absolute grey-values – i.e. depths – are affected by slight tilts of the chip caused by bending of the chip, or of the injection moulding mask. Another reason is that the chip can lie askew during the image acquisition due to dust particles on the locating surface. In order to remove this phenomenon from the raw data, the upper surface has to be estimated and subtracted using least squares plane fit [7].

Afterwards, the quality of the chip can be assessed by determining channel depths, identifying burrs, calculating transcription and detecting defects.

3.1 Determination of Channel Depth

The algorithm for determining the global mean channel depth is roughly divided into three steps. The results of the steps are depicted in Fig. 2.

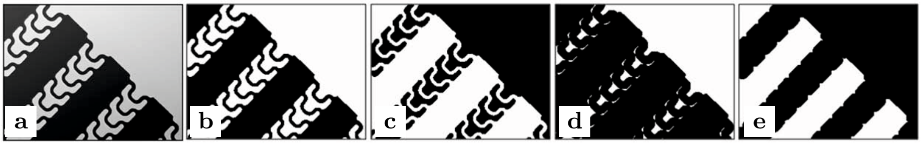


Fig. 2. The results of the intermediate steps of calculating the mean channel depth. **Left to right: (a)** Raw image – **(b)** Segmentation of upper surface – **(c)** Segmentation of channel layer – **(d)** Erosion of (b) – **(e)** Erosion of (c).

In a first step, the different layers have to be identified. To this end, Otsu segmentation is applied to the corresponding region in the reference image. This results in two or three segments depending on the present number of different layers.

Secondly, in order to remove unwanted areas, which are likely to contain burrs as well as measurement artefacts, morphological erosion is applied to the different segments. Otherwise, these areas would have a negative influence on the calculation and would distort the achieved results. Also small overlapping errors that may be caused by a possibly insufficient registration accuracy are eliminated.

Finally, the mean values r_i and the variances σ_i^2 of all regions in the sensed image, which correspond to the eroded segments derived from the reference image, are calculated. Thus, the mean channel depths d_i are calculated as the absolute difference between the mean value of the eroded upper surface and the mean values of the respective channel layers.

3.2 Identification of Burrs

The recognition of high burrs is realized by the application of several image processing techniques. The algorithm basically consists of three processing steps. The results of the subsequent steps are shown in Fig. 3.

In an initial step, regions have to be identified that are likely to contain burrs. Thus, regions near channel boundaries have to be selected. In contrast, regions on filter columns are not of interest. In order to remove them from analysis, a combination of Otsu segmentation, morphological erosion and region-growing segmentation is employed. The first segmentation step identifies the upper surface, which also contains filter columns. By choosing an appropriate size of the structure element, the applied morphological erosion entirely removes the pillars. As this erosion also moves the channel boundaries, the latter are reconstructed by a subsequent region-growing segmentation with the erosion result as seed points. After this "opening by reconstruction" procedure, the full upper surface without filter columns is available.

Secondly, in order to isolate regions that are likely to contain burrs, a morphological gradient (difference of dilation and erosion) is applied to the segmented upper surface. As a result, the potential burr regions are isolated. Dilation is

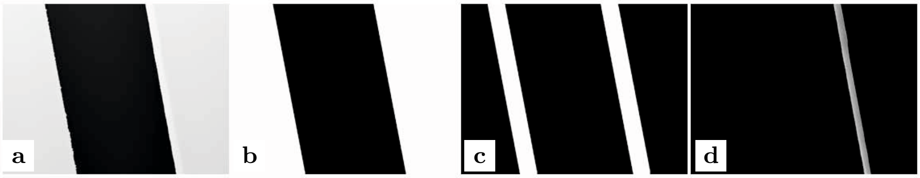


Fig. 3. Results of essential steps in the identification of high burrs. **Left to right: (a)** Raw image – **(b)** Segmentation of the upper surface – **(c)** Morphological gradient – **(d)** Identified burr.

applied to discard the effects of small misalignments caused by inaccuracies of the registration process.

Finally, in order to find high burrs, the identified regions are examined. Burr is signalled at those locations i where the measured height value z_i exceeds the average level $\bar{r}$ of the upper surface by more than a predefined threshold t, i.e. where $|\bar{r} - z| > t$. Burr locations are then collected into connected sets, and aggregated information on these burr regions is reported. This comprises the number of regions, the highest value of each region, the position of the burrs and statistics about sizes and ellipsoidal diameters.

3.3 Calculation of Transcription Accuracy

The algorithm for determining the transcription of all filter columns basically consists of four steps. The results of the steps are presented in Fig. 4.

First, the upper surface is segmented by the Otsu method. In order to identify the single filter columns on the entire upper surface, the surface is divided into its connected components by a morphological labeling step. All connected components are classified into filter columns or non-channel regions. This decision can simply be made based on the sizes of the identified objects. Note that the segmentation procedure is performed on the reference data so that defects of pillars or in channels will not introduce erroneous splitting or joining of segments.

To suppress influence from burrs in near-edge areas and measurement artifacts, all objects identified before are reduced in size by erosion.

Finally, a transcription parameter is calculated for each filter column. This involves the determination of the average value $\bar{r}_i$ of the overlapping regions between the eroded objects (columns) and the template image as well as the calculation of the average value of the upper surface $\bar{r}_0$. Thus, the transcription parameters are the height differences d_i between those respective values,

$$d_i = |\bar{r}_0 - \bar{r}_i| \, , \quad i = 1, \ldots, n \, . \tag{5}$$

3.4 Detection of Defective Regions

The algorithm is divided into three main steps: exclusion of biasing areas, detection of deviations and characterization of the detected deviations. The results of the steps are depicted in Fig. 5.

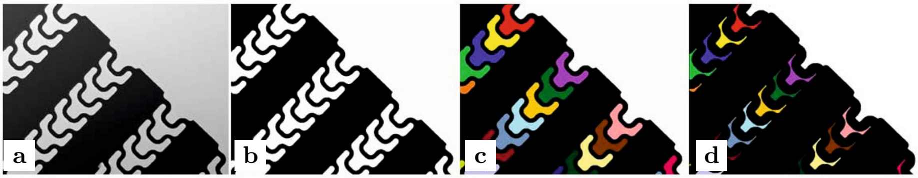


Fig. 4. Results of individual processing steps for calculating the transcription accuracy of all filter columns. **Left to right: (a)** Raw image – **(b)** Segmentation of the upper surface – **(c)** Result of morphological labeling – **(d)** Erosion of the labeled objects.

In a first step, all regions which do not contribute and may have a negative influence on the detection results are excluded from the inspection. As already explained above, those biasing areas are located around the channel boundaries. In order to identify them, the surfaces are segmented by the Otsu method. Similar as in the transcription accuracy analysis, a morphological gradient operation yields an image which contains a segmentation of all areas around the borders.

The second step is to detect all deviations between the reference and the transformed template image. By calculating the difference between the sensed image and the model image as well as excluding the overlapping regions of the result from the first step, it is possible to identify all real deviations. Further application of morphological opening removes the remaining artefacts. This corresponds to a similar approach proposed by Tait et al. [20]. The determination of the structure element's size plays a crucial role in this context. Oversized elements would result in the removal of possible defects, undersized would lead to the detection of small defects, which actually are phantoms. The defects or impurities are then located by applying labeling. At this point the defects are detected as well as labeled and can be characterized.

The last step aims at characterizing the individual defects and reporting information about them. For each defect, its ellipsoidal diameter and size are determined. Aggregating over all defects, statistics of these quantities are calculated, including mean, standard deviation, minimum and maximum. Secondly, information about the location of the defects is gathered. The locations of the defects are determined by several processing steps. At first a distinction between defects on the upper surface outside channels, and defects in channels is made. This is accomplished by inspecting the defects overlapping regions in the reference image and the corresponding intensity values in the model. Additionally, defects within channels are classified into defects within a filter structure, outside but near to a filter structure, or remote from all filter structures. This is accomplished by calculating Euclidean distances between a given defects and the nearest filter columns. If the distances of two or more columns are beneath a given threshold, the defect is located in a filter structure and has the potential to block the structure. If only one filter column is found in the proximity, the defect lies near the filter structure but not within it. Finally, if the number of nearby filter columns is equal to zero, the defect lies completely outside the filter structure.

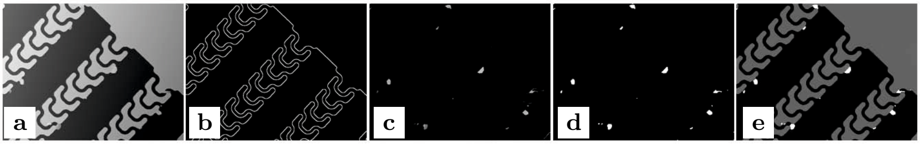


Fig. 5. Results of the main processing steps for defect detection. **Left to right: (a)** Raw image – **(b)** Morphological gradient – **(c)** Difference of model and acquired image – **(d)** Opening of (c) – **(e)** Chip image with detected defects overlaid in white (originally, defects are shown in color).

4 Experimental Validation

4.1 Comparison with Manual Inspection

A set of 42 test images were selected to calculate the defined quality parameters. In order to draw a comparison with manual inspection, the manually achieved results were gathered. Unfortunately, it turned out that transcription parameters could not be compared since they showed an excessive operator dependency in manual measurements. Nevertheless, channel depths and burrs could be compared. Results are shown in Fig. 6.

On average there is a difference of $(0.2 \pm 0.15)\,\mu\text{m}$ as far as channel depths are concerned. This deviation is not significant. However, the surface-based calculation of the channel depth is not only highly objective but also reflects the real channel depth of the whole channel depicted in the acquired image.

Surprisingly, there is no correlation between the automatically and the manually determined burrs. From the results in Fig. 6 it is evident that burr heights derived in the automatic procedure are systematically lower than the manually gauged burrs. Part of a possible explanation for this might be the definition of the burr itself. Burrs are raised edges which exceed the normal part geometry and occur at channel edges. Contrary to this definition, no distinction between edges on the filter columns and channel edges is made in the manual gauging process. Moreover, operators doing manual measurements tend to gauge burrs against the surface level in the vicinity of the burr region, which may involve a systematic underestimation of this surface level as compared to the globally adjusted surface level of the automatic procedure.

4.2 Defect Detection

In order to illustrate the localization capabilities of defect detection, the identified deviations were investigated. A set of 50 test images was used for this purpose. Tab. 4.2 shows the localization results. An overall number of 130 defects was detected by the system. At first, the defects were localized as either being on the upper surface or within the channel. Altogether, 92 defects were classified as channel defects. The remaining 38 defects were categorized as being located on the upper surface. In addition, the 92 channel defects were classified

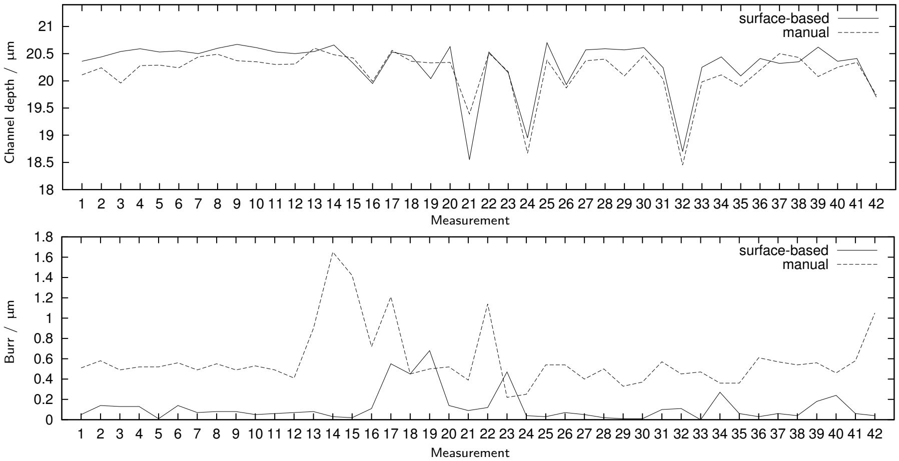


Fig. 6. Top to bottom: (a) Comparison of channel depths – **(b)** Comparison of burrs

into three groups: in the filter structure, near the filter structure and outside the filter structure. The system categorized 50 defects as being in the filter structure, 23 defects near the filter structure and 19 as being completely outside the filter. Additionally, the identified defects were manually inspected in order to check the reliability of the system. The classification was done independently from the automated localization. The decision between channel defects and surface defects completely matched with the automated inspection. However, results of the second classification scheme slightly differed between manual investigation and automated inspection. A total number of 54 defects were considered to be in the filter, 20 near the filter and 18 completely outside the filter structure. These small differences may indicate that the parameters for defect localization were still not optimally adapted. Further work has to be done on fine tuning of the parameters.

Table 1. Comparison of manual and automated localization of 130 detected defects

Localization	Number of defects	
	Automated	Manual
On Surface	38	38
In Channel	92	92
– In Filter	50	54
– Near Filter	23	20
– Outside Filter	19	18

5 Conclusion

We have demonstrated a system for automated quality control of microfluidic chips. The system is capable of both the calculation of different quality characteristics and detection as well as localization of defects. Image registration is used to localize the region of interest and establish independence of rotation and translation. Morphological image processing techniques play a crucial role in the analysis step.

Experimental comparison with manual inspection demonstrated the advantages of the system. The benefit is not only that the demand for human workforce for quality inspection is reduced but also the reproducibility and reliability of so computed quality parameters is higher due to the elimination of the subjective manual gauging. The removal of the human influence leads to an objective evaluation method.

Further work is necessary to raise the performance of the developed system, primarily by more efficient implementation of critical parts, and overall code optimization. Ongoing work focuses on revising the parameters and criteria in the quality inspection. Besides introducing new quality criteria, defect detection needs to be made more specific. For example, shape defects should be distinguished from surface defects. Shape irregularities may result in the jamming of filter structures, thus interfering with the chip's functionality. In contrast, surface defects in uncritical areas may be completely irrelevant. Considerably more work will be needed to develop classification capabilities. Given the determined quality parameters and the number, size as well as location of the detected defects, a classification system will be able to support the operators in deciding whether the chip should be accepted or rejected.

We expect that the further development of the presented system will even strengthen its benefit in terms of objectivity and reliability, and turn it into a powerful tool in routine quality inspection.

References

1. Asha, V., Bhajantri, N., Nagabhushan, P.: GLCM-based chi-square histogram distance for automatic detection of defects on patterned textures. International Journal of Computational Vision and Robotics 2(4), 302–313 (2011)
2. Chiou, Y.-C., Zhang, Y.-K.: An edge-based registration method for locating defects on PCB films. In: Arabnia, H.R. (ed.) Proc. of the 2006 International Conference on Image Processing, Computer Vision, and Pattern Recognition, June 26–29, vol. 1, pp. 209–215. CSREA Press, Las Vegas (2006)
3. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley, New York (2001)
4. Elbehiery, H., Hefnawy, A., Elewa, M.: Surface defects detection for ceramic tiles using image processing and morphological techniques. World Academy of Science, Engineering and Technology 5, 158–162 (2005)
5. Goshtasby, A.: Image Registration: Principles, Tools and Methods. Springer, New York (2012)

6. Goutsias, J., Heijmans, H.: Fundamenta morphologicae mathematicae. Fundamenta Informaticae 41, 1–31 (2000)
7. Eberly, D.: Least Squares Fitting of Data, `http://www.geometrictools.com` (last visited December 01, 2012)
8. Ibrahim, I., Ibrahim, Z., Khalil, K., Mokji, M.M., Abu Bakar, S.A.R.S., Mokhtar, N., Ahmad, W.K.W.: An improved defect classification algorithm for six printing defects and its implementation on real printed circuit board images. International Journal of Innovative Computing, Information and Control (5(A))(2-12), 3239–3250
9. Leta, F., Feliciano, F., Martins, F.: Computer vision system for printed circuit board inspection. ABCM Symposium Series in Mechatronics 3, 623–632 (2008)
10. Li, S.: Markov Random Field Modeling in Image Analysis. Springer, London (2001)
11. Lin, H., Chiu, Y., Hsu, S.: A visual inspection system for quality control of optical lenses. International Journal of the Physical Sciences 6(11), 2701–2709 (2011)
12. Mandelbrot, B.: The Fractal Geometry of Nature. W.H. Freeman, New York (1983)
13. Maintz, J., Viergever, M.: A survey of medical image registration. Medical Image Analysis 2, 1–36 (1998)
14. Mital, A., Govindaraju, M., Subramani, B.: A comparison between manual and hybrid methods in parts inspection. Integrated Manufacturing Systems 9, 344–349 (1998)
15. Moganti, M., Ercal, F., Dagli, C.H., Tsunekawa, S.: Automatic PCB inspection algorithms: a survey. Computer Vision and Image Understanding 63, 287–313 (1996)
16. Ng, H.: Automatic thresholding for defect detection. Pattern Recognition Letters 27, 1644–1649 (2006)
17. Novak, I., Hocenski, Z.: Texture feature extraction for a visual inspection of ceramic tiles. Proceedings of ISIE 3, 1279–1283 (2005)
18. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics 9(1), 62–66 (1979)
19. Shih, F.: Image Processing and Pattern Recognition. Wiley, New York (2010)
20. Tait, R., Schaefer, G., Nolle, L.: Automated visual inspection using a distributed blackboard architecture. International Journal of Simulation, Man and Cybernetics 7(3), 12–20 (2006)
21. Tsai, D., Lin, C., Huang, K.: Defect detection in coloured texture surfaces using Gabor filters. Imaging Science Journal 53(1), 27–37 (2005)
22. Vilnrotter, F., Nevatia, R., Price, K.: Structural analysis of natural textures. IEEE Transactions on Pattern Analysis and Machine Intelligence 8, 76–89 (1986)
23. Wiltschi, K., Pinz, A., Lindeberg, T.: An automatic assessment scheme for steel quality inspection. Machine Vision and Applications 12, 113–128 (2000)
24. Xie, X.: A review of recent advances in surface defect detection using texture analysis techniques. Electronic Letters on Computer Vision and Image Analysis 7(3), 1–22 (2008)
25. Xie, X., Mirmehdi, M.: Texture exemplars for defect detection on random textures. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) ICAPR 2005. LNCS, vol. 3687, pp. 404–413. Springer, Heidelberg (2005)
26. Zheng, H., Kong, L., Nahavandi, S.: Automatic inspection of metallic surface defects using genetic algorithms. Journal of Materials Processing Technology 5, 158–162 (2002)
27. Zitova, B., Flusser, J.: Image registration methods: a survey. Image and Vision Computing 21, 977–1000 (2003)

Geography, Mathematics and Mathematical Morphology

Christine Voiron-Canicio

ESPACE, University of Nice – Sophia Antipolis - CNRS, France

Abstract. Mathematical Morphology (MM) has been introduced in geographical sciences during the years 1970-1980. However it did not find the same echo in the geographer community according the areas of research. Unlike remote sensing where MM tools have been used as early as in the eighties and are nowadays widespread, in the research works resorting to spatial analysis and modelling, MM is much rarer. And yet morphological analyses exactly match the purpose of spatial analysis. This talk aims to demonstrate the relevance of MM in geography and more precisely in spatial analysis. The three applications proposed focus on socio-economic issues: urban zones of influence detection, regional differentiations analysis and spatial modelling. Finally, are highlighted and discussed the major shortcomings which hold up the spread of MM in geography, planning and geomatics.

Keywords: Geography, GIS, Morphology, Modelling, Spatial Analysis.

1 Introduction

Geography as it is being practiced nowadays is little known not only by the public at large, but also by other scientists and yet it very often works in collaboration with them. This paradox is mainly due to the fact that there are numerous currents in geography, some of them opposed, which contributes to blurring its image. On the other hand, because of its interdisciplinary character, it is perceived as a discipline at the crossroads of various sciences. Yet the object of geography is clearly defined – it is neither man nor nature, instead it studies the spaces occupied by man and nature. Its purpose is to explain the relationships between societies and their space, and geographical currents correspond to the different points of view adopted to achieve this.

The thoughts expressed in this paper are those of a geographer belonging to the quantitative geography current. This expression, which is not really the best, describes a way of analysing geographical phenomena using mathematics. Mathematical geographic analysis was developed in two phases. The first dates back to the introduction of statistical analysis and descriptive and explanatory methods in the course of the years 1950-1960; then at the beginning of the seventies, mathematical analysis became integrated into geographic analysis and followed the development of theorisation and modelling. Geographers then endeavoured to relate mathematical concepts with the geographic concepts they wanted them to represent.

One central notion in geography is the notion of spatial interaction – a reciprocal action between two or more places. Gravity models, by analogy with Newton's law of

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 520–530, 2013.

universal gravity, were then widely used to formalise, study and predict the geography of flows and interactions. The distribution of interactions in an ensemble of places is considered as dependent on their configuration, i.e. the force of attraction of each one of them and the difficulty of communication between them. The spatial interaction functions most used to describe the influence of distance are negative power functions (called Pareto functions) and negative exponential functions. There are various types of gravity models. Potential models are based on measuring the accessibility of the place and assessing the amount of opportunities according to the position of the place. The aim of position models, for example Reilly's and Huff's, is to determine the theoretical market areas of an ensemble of places [1-2].

More recently, quantitative geographers turned their interest towards morphological theories – R. Thom's Catastrophe theory, B. Mandelbrot's Fractals theory, I. Prigogine's Dissipative Structures theory. Some have, more than others, generated research work. Fractals induced the most research work. The fractal dimension was used to test the existence of a hierarchical organisation in urban morphology and population distribution [3-5].

Intriguingly, Mathematical Morphology (MM) did not find the same echo in the geographer community. However, this observation should be qualified by distinguishing two areas of research: geosciences and remote sensing on one hand, spatial analysis applied to regional and urban analyses on the other hand. As regards remote sensing, MM tools have been used as early as in the eighties and are nowadays widespread [6-7]. On the other hand, in the second area, resorting to MM both in spatial analysis and modelling is much rarer, though the first applications in regional geography date back to 1989 [8]. Research works in spatial analysis are mainly carried out by researchers of the Laboratory ESPACE, a CNRS research unit. And yet mathematical morphology's method exactly matches the purpose of spatial analysis.

The potential of MM in spatial analysis will be demonstrated by focusing on socio-economic issues: urban zones of influence detection, regional differentiations analysis and spatial modelling. Finally, are highlighted and discussed the major shortcomings which hold up the spread of MM in geography, planning and geomatics.

2 The Relevance of MM in Spatial Analysis

The first law of geography according to Waldo Tobler is « Everything is related to everything else, but near things are more related than distant things » [9]. The concept of spatial dependence that forms the foundation of spatial analysis follows from that law.

Too often, spatial analysis is reduced to a set of methods and tools measuring the distribution of phenomena across space, such as spatial autocorrelation, interpolation and concentration or dispersion indices, for example. Integrating these techniques in Geographical Information System (GIS)-type software has made their use commonplace, but also contributed to mixing up spatial analyses and geomatics. Yet, Spatial Analysis is first a way of viewing geography from a spatial standpoint. This position consists in taking explicitly the characters of the space into account – distance, spacing, neighbourhoods – in order to determine in which way they participate in the organisation of space. In concrete terms, the "spatial standpoint" consists in:

- Paying more attention to the place geographic objects occupy in the space than to their number. The notions of spacing and neighbourhood are at the core of spatial analysis.
- Giving preference to conditional analysis over global analysis and trying to find out how a geographic phenomenon spreads according to its distance from another.
- Finding out the places'relative weights. What counts is not so much the absolute value of a variable in relation to a place than the weight of that place in the surrounding space and its relationship with neighbouring places.
- Measuring the role played by the spatial environment, 2D and 3D, more or less close, in the dynamics of the territorial system being studied, and the "range" of interactions (the expanse of the area under influence, consequences in terms of speculation or vulnerability, or spatial effects and their territorial inequalities).
- Focusing on the way in which certain mechanisms are slowed down, accelerated or modified by particular spatiotemporal situations.

Mathematical Morphology is particularly adapted to this perspective and provides a formal frame to the morphological approach, both in spatial analysis and in modelling.

In spatial analysis, image analysis through MM differs from that used in remote sensing. The processing aims neither to recognize objects from visible forms, nor to quantify parameters, instead it aims to detect structures from known geographic objects. The idea is to make them explicit by means of a form that reveals the invisible relations forged between objects – which "gives form" to the structure being looked for.

Unlike in remote sensing, working images are not limited to satellite or aerial photographs. The document par excellence for spatial analysis using MM is the map - whether localisation maps, land surveys (investigations, excavations etc.) or zoning maps from GIS layers. The specificity of this kind of analysis lies in the fact that the images processed concern elements which are not visible directly: population data, socio-economic data, informations on behaviours, practices and standpoints. Mapped information also comes from prior statistical or mathematical processing: rate of population growth, of housing growth, urban pressure indices calculated on a communal scale, for example. It may also focus on zonings relating to the level of reliability of the spatial information collected (quality of archaeological prospecting, for example).

The working images are built using a spatialised data table. This kind of table is obtained by gridmapping the area being studied. The columns and lines correspond respectively to the x and y coordinates of the area and each cell contains information on the corresponding portion of space.

Themes and problems are most varied: detecting boundary effects, alignments, archaeological structures, analysing relations between urban fabric and road, space-time variations, detecting the hierarchical organisation of regional differentiations, the discontinuity between two regional subsets etc. [10-13].

In the following sections, three types of image processing representative of the usefulness of MM in spatial analysis and modelling are presented.

3 Applications

3.1 Detecting and Mapping Regional Contrasts

The first application aims to detect the home-building contrasts, in two departments of the South of France subjected to a strong residential pressure on the coastal areas which is one of the main features of the littoralisation process [14]. As disaggregated data on urbanisation are not available to make a retrospective analysis we work on the average annual growth rates of home-building aggregated at the municipal level (the commune).

Detecting spatial regionalisations consists in bringing out spatial discontinuities, in other words, gaps which appear along the boundaries of a territorial grid. This question can be treated within two different frameworks, statistical analysis and MM analysis. First, the degree of dissimilarity of contiguous spatial units taken by two is measured by a dissimilarity function. Its simplest form, i.e. the gap's absolute value, is chosen. Therefore, the dissimilarity function measures, for each pair of communes, the number of points between their respective rates of home-building. Only the highest dissimilarity values are represented on a map, enabling us to observe the location of the strongest discontinuities in the studied area. We chose to map the two last deciles of the discontinuity values (Fig. 1). The strongest local differentiations are shown in thick lines, and those who are a little less strong in medium-sized lines. As previously, the map of intra-regional differentiations is dependent on the class thresholds used. However, this technique can bring out neither existing regionalisations nor the way they fit into one another.

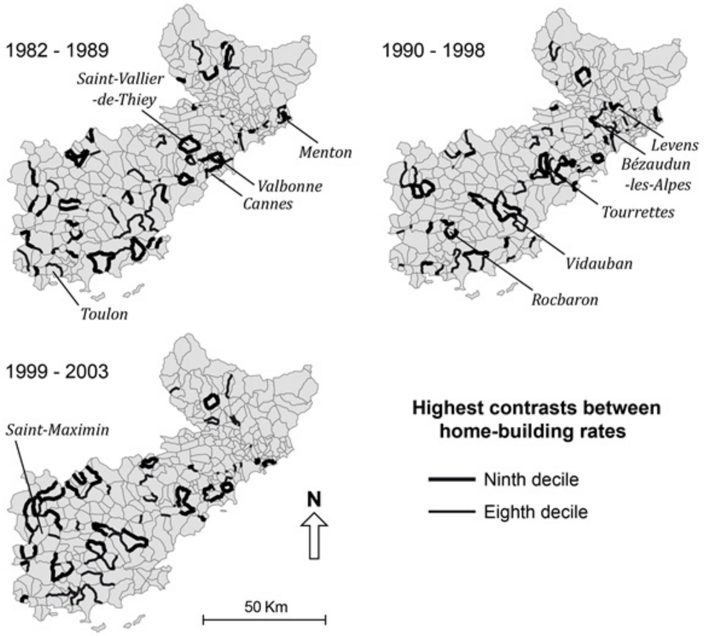


Fig. 1. Average annual growth rates of home-building on the French Riviera

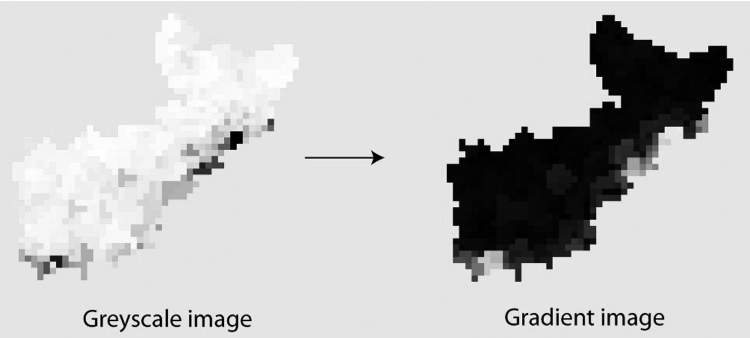


Fig. 2. Home-building images on the French Riviera (Alpes-Maritimes and Var)

Second, the morphological gradient is used to detect spatial differentiations of home-building values**.** The original image is converted into a mosaic image, where component contours are more highly contrasted (Fig. 2).

Once the morphological gradient has been calculated the gradient-mosaic image is segmented using the watershed transformation. The dividing lines delimiting the catchment basins are more and less high, according to the gradient values. The next process consists in removing these "walls" from the smallest to the highest in order to gradually suppress the internal differentiations [15]. This hierarchical pyramid method provides a convenient tool to detect the main regional structures in terms of contrasts. The Figure 3 represents successive hierarchy levels at different periods. The maps focus on the final results for each period.

This analysis enhances the characterisation of the littoralisation phases: first the individuation of some parts of the coast, then the formation of a continuous coastal strip and its progressive spread towards the inner lands. More complex differentiation processes appear then with the individuation of sub-regional systems connecting different parts of the coast with their hinterlands, and finally a homogenisation of hinterlands.

The gradient's hierarchical organisation through the pyramid transformation takes all existing levels of contrast into account. Unlike the image of the dissimilarity function the watershed transformation on the gradient image, provides homogeneous zones with closed outlines, instead of portions of outlines. It becomes possible to map in turn the regions corresponding to all levels of contrast and the specificities of these regions can be highlighting by crossing their perimeters with other data, for example land use data.

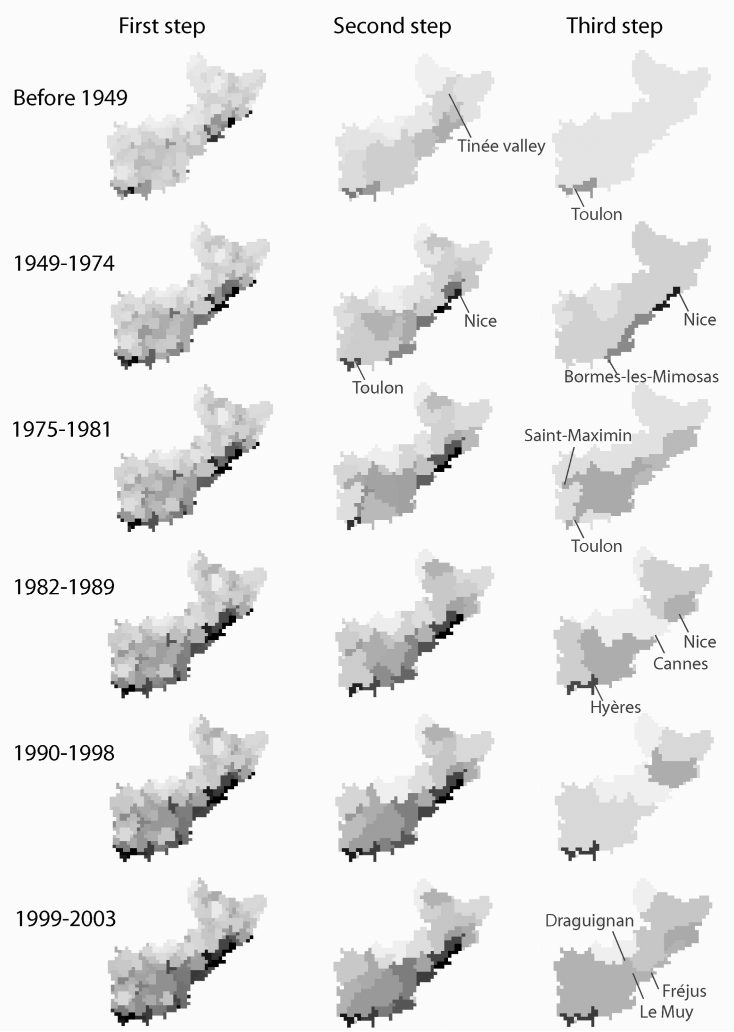


Fig. 3. Successive levels of hierarchy in home-building values on the French Riviera

3.2 Selecting Regional Centers and Detecting Their Influence Zones

In the second application, the watershed of a gradient image is also used but now the goal is to detect places which play a local or a regional part, starting from two grey-tone images [15]. The first one represents the population value of the 343 communes of Hérault, a department of the South of France; the second, the number of basic public or private services of each commune, (nursery, elementary and secondary schools, post office, bank, doctor, chemist, lawyer). The first stage consists in detecting the local minima on the basic services image. In order to range the 36 detected minima into hierarchical groups the usual approach generally considers their levels of services and their population values; here the chosen method is based on the levels of the morphological gradient.

Figure 4 summarises the steps of the image processing. The levels of the gradient are progressively removed, starting from the lowest (A). At each step the minima of the services image (B) are intersected with the remaining gradient levels. Image C displays the pixels belonging to both images and from which the minima are reconstructed (D). The speed at which the number of centers is decreasing after removing a gradient level, provides information about their significance. Some minima disappear very quickly; they correspond to 16 centers which stand out weakly against the neighbourhood. On the contrary 20 minima correspond to higher gradient levels. They can be divided into three groups. The following step will focus on them. By using the minima of each group separately, the marked watershed transformation on the population values is done.

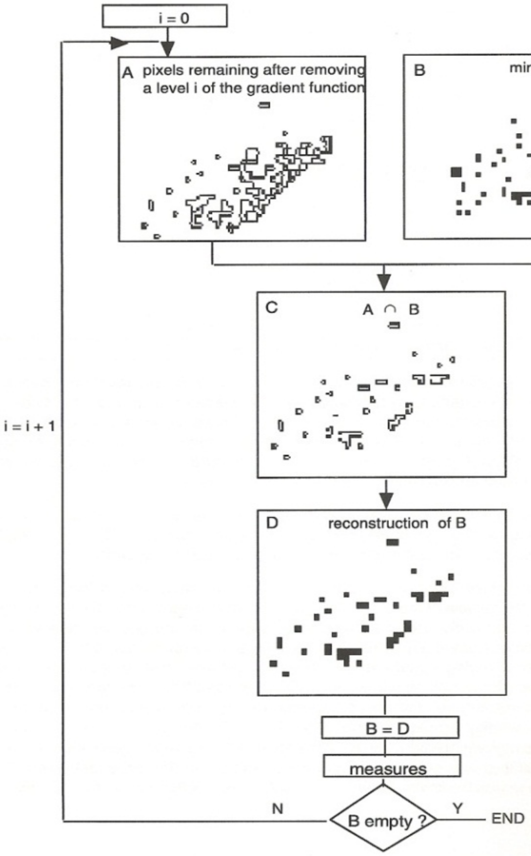


Fig. 4. Chart's flow of the classification of centers

All the limits of the nested regions have been drawn on a same image (Fig.5). The limits of the 20 catchment basins have been coloured with different grey tones. Finally, these theoretical influence zones have been compared with attraction areas provided by surveys about the travels from each commune to the most frequented city of the region [16]. The resemblance is quite striking. This application demonstrates that spatial criteria for selecting centers combined to segmentation and hierarchical pyramids provide limits quite similar to those obtained with statistical surveys. Moreover the morphological approach brings higher information since it makes it possible to join every small area of influence to the upper one and to display the set of nested spheres of influence.

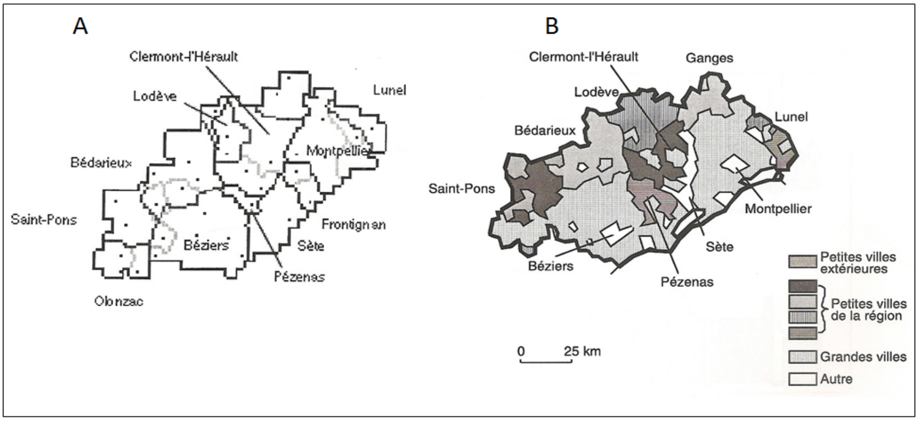


Fig. 5. Comparison between nested zones of influence (A) and map of attraction areas (B)

3.3 Simulating the Spread of Built-Up Areas

We will now present an example of spatio-morphological modelling applied to simulate the built-up areas spread on a coastal region of the Languedoc (Southern of France). In this approach, the spatial spread process depends on both proximity and morphology of the built-up areas. These features explain the "spatio-morphological" qualifier given to this model [17]. The model is carried out with data relating to 1977 and 1990. Five images are needed: the regional field on which the spread occurs; the state of the built-up areas surfaces in 1977 and in 1990; the roads and the extent of the protected natural zones.

Closings of increasing sizes are used to simulate the spread during the period of 1977-1990. The steps of the modelling by image processing have been summarised on the Figure 6. Each step of simulation consists of closing the built-up set (Image A1). The closing is conditional, i.e the closed set (image B) is intersected with the permitted region in the field, in order to eliminate pixels falling into the sea, into the ponds or into zones where urbanisation is not allowed. Such a conditional closing is repeated. At each step, the new built-up areas predicted (image C) are compared with the new built-up areas observed in 1990 (image A2). The criterion for stopping the spread process is the value of a coefficient of similarity, simil, calculated on the new built-up areas only.

simil (A2,C) = surface of intersection (A2,C) / surface of union (A2,C) (1)

d(A2,C) = surface of union (A2,C) - surface of intersection (A2,C) = [1-simil (A2,C)] * surface of union (A2,C) (2)

The coefficient simil is related to the distance d but unlike it, simil doesn't depend on the size of the images. We outline that simil (A2,C) is different of simil ($A2^c$, C^c). Its value is 0 when distance d is maximum and 1 when d is equal to zero.

The matching is performed for all probable sizes of closing and one takes that which maximizes the coefficient simil.

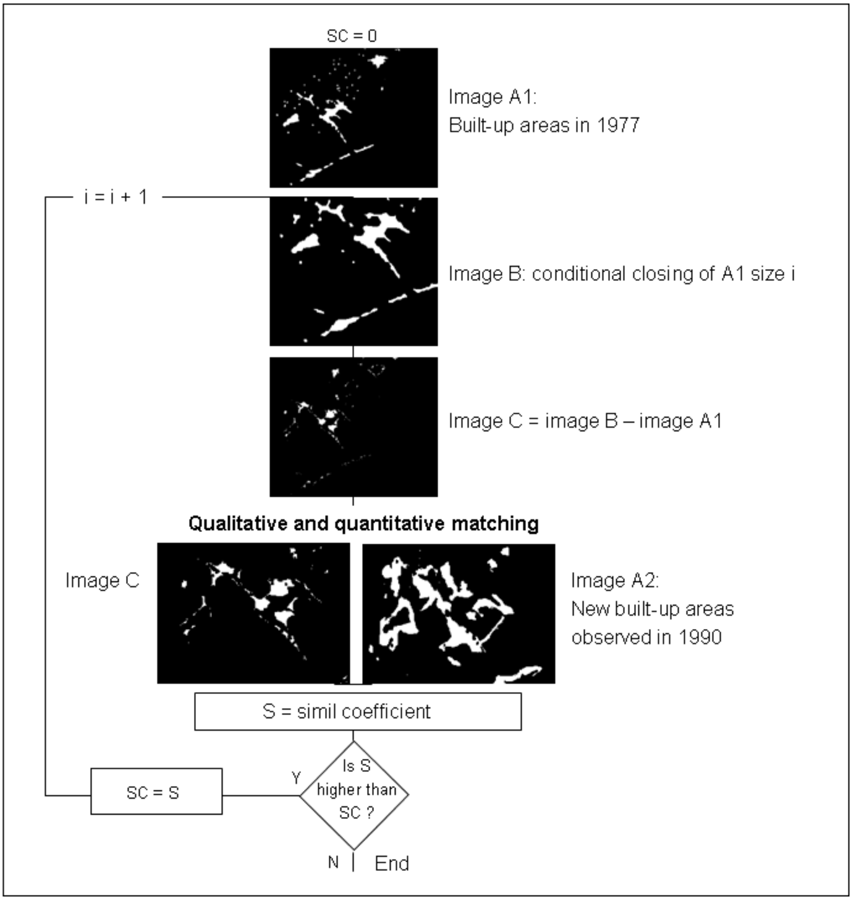


Fig. 6. The steps of the spatio-morphological modeling

Generally, map comparisons of observed and predicted states are used to evaluate the outputs of spatial models. Traditional pixel-by-pixel comparisons involve overlaying mappings to evaluate the similarity between two maps. However, the variability at the pixel scale is always high and we agree with the fact that the predictive models are not expected to be accurate at the pixel scale. They are expected to predict the approximate shapes and locations of the phenomenon under study [18]. Therefore, the similarity has been evaluated by taking into account a margin of error of three pixels. The step consists of dilating the new predicted surfaces by 1, 2 or 3 pixels successively, before performing the intersection with the new observed surfaces.

The output image of the intersection between predicted and observed new surfaces gives information about the built-up areas correctly located by the predictive model and makes it possible to detect both regions of agreement and of disagreement. Generally, this visual comparison is very useful because it contributes to improving the model by giving rise to new spread hypothesis.

4 Conclusion

MM offers huge application potential in geography, as much for structure analysis as for spatial modelling. Nowadays, spatial geo-archaeology is opening itself to MM, and new applications concern the processing of imprecise and uncertain geographic data by means of fuzzy sets and MM. However, this research concerns only a small number of initiated persons.

The spread of MM in the areas of geography, planning and geomatics is at present held up by technical and conceptual reasons. MM belongs to the image analysis which comes under remote sensing and there is a compartmentalization between remote sensing work and research work on territorial systems carried out in other environments: statistics, geomatics, cellular automata and multi-agent system. Nowadays, GISs have become basic tools for geography processing, yet most do not include mathematical morphology modules, except for a few elemental morphological transformations. Images have to be exported towards MM software, and then the processed images are reintroduced in the GISs.

Two majors have to be underlined, the lack of user-friendly dedicated software. On that matter, one can but regret the fact that the Micromorph educational software has been dropped. The same phenomenon occurred in the nineties with expert system software for the general public. The absence of MM module in GISs is also a big disadvantage. While numerous complementary modules are available, there is none on MM. Applications are at present too few for companies to include them in their products, but as a reverse effect, the absence of MM module prevents it from becoming widespread.

Then, MM must be taught in order to be correctly used and developed in geography; yet, very few university programs include MM in the teaching of theories and methods of spatial analysis.

Added to these technical reasons is the fact that space is more perceived as a field of forces than a field of forms, explaining geographic phenomena by fields of forces is still prevailing in modelling.

References

1. Huff, D.L.: Defining and Estimating a Trade Area. Journal of Marketing 28, 34–38 (1964)
2. Reilly, W.J.: The Law of Retail Gravitation. Knickerbrocker Press, New York (1931)
3. Frankhauser, P.: The fractal approach. A new tool for the spatial analysis of urban agglomerations. Population. Special issue New methodological Approaches in the Social Sciences, 205–240 (1998)
4. Genre-Grandpierre, C.: La desserte spatiale des réseaux de transport routier: une approche fractale. Flux 38, 56–68 (1999)
5. Tannier, C., Thomas, I., Vuidel, G., Frankhauser, P.: A fractal approach to identifying urban boundaries. Geographical Analysis 43, 211–227 (2011)
6. Destival, I.: Mathematical morphology applied to remote sensing. Acta Astronautica 13, 371–385 (1986)
7. Gallice-Matti, C.: La télédétection pour l'analyse spatiale : application aux espaces périurbains de la région urbaine de Lyon. Lyon 3 University. Thesis (2005)
8. Voiron-Canicio, C.: Les départements de France redessinés. Mappemonde 4, 26–28 (1989)
9. Tobler, W.: A computer movie simulating urban growth in the Detroit region. Economic Geography 46(2), 234–240 (1970)
10. Voiron-Canicio, C.: Analyse spatiale et analyse d'images. GIP Reclus, collection Espaces modes d'emploi, Montpellier (1995)
11. Voiron-Canicio, C., Doveri, E., Fusco, J.: Spatial archaeology using image analysis and mathematical morphology (2011), http://halshs.archives-ouvertes.fr/halshs-00778526
12. Maignant, G.: Dispersion de polluants et morphologie urbaine. L'Espace Géographique 2, 141–154 (2007)
13. Saint-Amand, P.: L'adéquation d'un système de transport aux systèmes territoriaux méditerranéens: pour une mobilité durable. Modélisations et aide à la décision. Thesis.University of Nice (2010), http://tel.archives-ouvertes.fr/tel-00565919
14. Liziard, S., Voiron-Canicio, C.: The Contribution of Mathematical Morphology in Spatial Analysis of Aggregated Data: Home-Building Evolution in the French Riviera during the Twentieth Century. In: 15th AGILE International Conference on Geographic Information Science, http://agile.gis.geo.tudresden.de/web/index.php/conference/proceedings/proceedings-2012
15. Voiron-Canicio, C.: Selecting places and detecting nested zones of influence by using image analysis. In: VIIth Coloquio de Geografia Cuantitativa, modelos y sistemas de informacion en geografica, Vitoria, pp. 348–356 (1996)
16. Brunet, R.: Atlas Permanent de la Région Languedoc Roussillon (1990)
17. Voiron-Canicio, C.: Predicting the Urban Spread Using Spatio-Morphological Models. In: Murgante, B., Borruso, G., Lapucci, A. (eds.) Geocomputation & Urban Planning. SCI, vol. 176, pp. 223–236. Springer, Heidelberg (2009)
18. Power, C., Simms, A., White, R.: Hierarchical Fuzzy Pattern Matching for the Regional Comparison of Land Use Maps. International Journal of GIS 15, 77–100 (2001)

Author Index