This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

A morphological approach to curvature-based evolution of curves and surfaces

Pablo Márquez-Neila, Luis Baumela and Luis Alvarez

Abstract—We introduce new results connecting differential and morphological operators that provide a formal and theoretically grounded approach for stable and fast contour evolution. Contour evolution algorithms have been extensively used for boundary detection and tracking in computer vision. The standard solution based on partial differential equations and level-sets requires the use of numerical methods of integration that are costly computationally and may have stability issues. We present a morphological approach to contour evolution based on a new curvature morphological operator valid for surfaces of any dimension. We approximate the numerical solution of the curve evolution PDE by the successive application of a set of morphological operators defined on a binary level-set and with equivalent infinitesimal behavior. These operators are very fast, do not suffer numerical stability issues and do not degrade the level set function, so there is no need of re-initializing it. Moreover, their implementation is much easier since they do not require the use of sophisticated numerical algorithms. We validate the approach providing a morphological implementation of the *Geodesic Active Contours*, the *Active Contours Without Borders* and *Turpopixels*. In the experiments conducted the morphological implementations converge to solutions equivalent to those achieved by traditional numerical solutions, but with significant gains in simplicity, speed and stability.

Index Terms—Computer vision, Mathematical Morphology, Curve Evolution, Level-Sets, Morphological Snakes

1 Introduction

Active contours or snakes are one of the most widely used computer vision tools [1], [2]. Although they provide a unified account of a number of visual problems, including detection of edge and subjective contours [2] and stereo matching [3], they have been extensively used for object boundary detection and tracking [1], [2], [4], [5], [6], [7], [8], [9] as well as segmenting 2D [10], [11], [12], [13], [14] and tensor images [15]. Recent results have shown that they can achieve robust tracking performance over long and challenging sequences with dramatic changes in target shape and appearance [16], [17] as well as overlaps, partial occlusions and poor image contrast [18]. It has also been used for oversegmentation [19]. These tasks are formulated in variational terms, where an image induces an energy functional on a curve or surface. Minimizing the functional in a steepest descent manner evolves the surface towards a local minimum that represents the solution of the problem.

Despite its great success, the original parametric active contour approach depends on the parametrization of the contour and cannot naturally handle changes in the topology of the curve. These issues were addressed in subsequent approaches such as the *Geodesic Active Contour* (GAC) [20], [21] and the *Active Contours Without Edges* (ACWE) [10], [11]. In the GAC the energy functional is a geodesic in a Riemannian manifold with a metric induced by image features, in its simplest case, the target borders. The ACWE does not need well defined borders and it is less sensitive to the initial configuration and to the model parameters. Both approaches are based on the level-set formulation [22], [23]. In this case the curve is evolved by propagating an interface represented by the zero level-set of a smooth function, using a time-dependent partial differential equation (PDE). The solution to this PDE is costly computationally, and in the case of the simplest finite-difference explicit numerical scheme, it has stability constraints on the size of the time step. Absolutely stable solutions to the GAC model improve the stability by combining a semi-implicit discretization with an additive operator splitting (AOS) [24], [25]. Level-set solutions typically develop steep or flat gradients that yield inaccuracies in the numerical approximation [26]. This is usually solved by periodically re-initializing the level-set function as a distance to the zero level-set, which can also be addressed as a front propagation problem [27]. This again increases the computational cost of the method and reduces the topological flexibility, since it prevents the level-set from creating new contours far away from the initial interface [26]. For the ACWE model, however, this re-initiliazation is optional [10].

Both the stability constraints and the necessity of re-initializing the distance function render traditional level-sets approaches as problematic schemes in time-

- *Pablo Márquez-Neila is with the Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Spain.*
E-mail: p.mneila@upm.es
- *Luis Baumela is with the Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Spain.*
E-mail: lbaumela@fi.upm.es
- *Luis Alvarez is with the Departamento de Informática y Sistemas, Universidad de las Palmas de Gran Canaria, Spain.*
E-mail: lalvarez@dis.ulpgc.es

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

critical applications. A considerable number of studies attempt to alleviate the computational demand reducing the domain of computation in a narrow band around the zero level-set together with multi-scale techniques, e.g. [25]. Many of the most efficient approaches [9], [12], [28], [29] avoid directly solving the PDE. They are based on two key ideas. The curve is implicitly represented by the set of image points neighboring the target interface. It is evolved by successive introduction and elimination of points in the interface set. Geometric properties such as curvature and normal directions to the curve are approximated using local operations on the set of interface points.

Other approaches to contour evolution try to find globally optimal solutions. This is possible for the GAC model when additional constraints are imposed. For example, Cohen et al. [30] use as additional constraints the locations of the curve end points. The method from Appleton et al. [31] requires a single point known to be contained in the area inside the curve. Other recent works find a convexification of the energy functional via functional lifting and convex relaxation [32], [33], [34]. The minima of the resulting convex functional coincides with the global minima. The minimization is based on the duality of the total variation norm, which is much faster than the traditional optimizations based on the Euler-Lagrange equations and it is not sensitive to the singularities of the level set function. Bresson et al. [32] introduce and globally minimize three new functionals, one closely related to the ACWE model. Their minimization, based on the duality of the TV-norm, is reported to be up to 60 times faster than the traditional solution obtained with the Euler-Lagrange equation.

Despite the additional assumtions that some of the global methods require, the advantages of a globally optimal solution are unquestionable. However, approaches seeking local minima are also of interest in applications that require an intrisically local solution, such as for example for tracking deformable objects [16], [17] or for oversegmentation [19]. Moreover, in complex image segmentation problems it might be very difficult to find a set of image features and an energy functional whose global optimum is the desired segmentation. This is the case, for example, in the analysis of Electron Microscopy (EM) images [35], for which user interaction in conjunction with a local approach is presently the best available approximation.

Our main goal in this work is to provide a formal, theoretically grounded, approach for stable and fast local contour evolution. We base our framework in the mathematical morphology. We substitute the terms that appear in the PDEs of contour evolution algorithms for morphological operators that have equivalent infinitesimal behavior. Then, the numerical solution of the PDE is approximated by the successive application of morphological operators. The level-set surface is now much simpler to define. We assign a value of 0 outside the contours and 1 inside. These operators are very fast, do not suffer numerical stability issues and do not degrade the level set function, so no re-initialization is required. Moreover, their implementation is much easier since they do not require the use of sophisticated numerical algorithms.

To formally support our solution we introduce new results that relate differential and morphological operators. The connection between differential and morphological operators has also been studied before. Lax was the first to use multi-scale dilations and erosions to give stable and efficient numerical schemes for solving PDEs [36]. Later it was rediscovered by several authors [37], [38]. Now, it is well-known that the PDE evolution rule which describes a curve moving along its normal behaves like the morphological operators dilation and erosion acting on the level set function [39]. However, not all PDEs have equivalent morphological operators. The most important PDE contour evolution rule, the mean curvature evolution [39], [40], has no known morphological equivalent operator. The importance of curvature-based evolution lies on the fact that it commonly appears in most PDE-based algorithms as a regularizing term. Hence the interest in finding a morphological equivalent. In this direction, some advances were achieved by Catté, Dibos and Koepfler proving that the mean curvature evolution for planar curves can be replaced by the mean of two morphological operators [41].

Our work has several contributions. First, we introduce a curvature morphological operator that can be used for curve evolution. Second, we prove that our operator can be generalized to higher dimensions and, therefore, it can be used for the evolution of curves, surfaces and hyper-surfaces of any dimension. Third, we show how the composition of different morphological operators approximates the numerical solution of PDEs for hyper-surface evolution, with significant gains in simplicity, speed, and stability. Specifically, we introduce morphological versions of the *Turbopixels* supersegmentation algorithm and two of the most popular curve evolution algorithms, the *Geodesic Active Contours* (GAC) [20] and Chan and Vese's *Active Contours Without Edges* (ACWE) [10].

In [42] we presented a morphological approach to the evolution of 2D contours based on which we introduced the Morphological GAC. Here we generalize this result to surfaces of any dimension, introduce the *Morphological Active Contours Without Edges* (Morphological ACWE), the *Morphological Turbopixels* and perform a larger set of experiments.

The rest of the paper is organized as follows. In Section 2 we present background knowledge about differential equations and morphological operators. Section 3 introduces the curvature morphological operator and studies its behavior in 2D, 3D and n-dimensional cases. We introduce the *Morphological Snakes* model in Section 4. Some implementation details are given in Section 5. Finally, we perform experimental analysis and draw conclusions in Sections 6 and 7.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

2 Background

In this section we briefly review both explicit and implicit curve and surface evolution, morphological operators and the relation between PDEs and morphological operators. The interested reader may consult [39] for §2.1, [40] for §2.2 and [41] for the last part of §2.3.

2.1 PDEs and contour evolution

Differential operators are fundamental tools in the computer vision and graphics communities. They are used in a variety of tasks such as determining the length of geodesic curves and computing the curvature of surfaces or smoothing meshes. Differential operators are key to contour evolution with PDEs [39], [40], where the infinitesimal change of a contour is given by a differential operator. We begin by recalling some information about curve evolution.

Let $\mathcal{C} : \mathbb{R}^+ \times [0,1] \to \mathbb{R}^2 : (t,p) \to \mathcal{C}(t,p)$ be a parametrized 2D curve over time. A differential operator L defines the curve evolution with the PDE $\mathcal{C}_t = L(\mathcal{C})$. Different forms of L result in different types of evolution. It is not difficult to see that every L can be rewritten as $L(\mathcal{C}) = \mathcal{F} \cdot \mathcal{N}$, where $\mathcal{N}$ is the normal to the curve and $\mathcal{F}$ is a scalar field —possibly depending on the curve— which determines the velocity of evolution of each point in the curve.

Out of all the possible forms that L can take, a few are of special interest for their theoretical properties and because they have been extensively used. First, L may be defined as the normal to the curve (i.e., $\mathcal{F} \in \{1, -1\}$), $\mathcal{C}_t = \mathcal{N}$ or $\mathcal{C}_t = -\mathcal{N}$. Here, the curve moves along its normal direction with constant velocity. Second, if $L(\mathcal{C}) = \mathcal{K}\mathcal{N}$ (i.e., $\mathcal{F} = \mathcal{K}$), we get the *intrinsic heat equation* [40], $\mathcal{C}_t = \mathcal{K}\mathcal{N}$, where $\mathcal{K}$ is the Euclidean curvature of $\mathcal{C}$. The curvature flow given by this expression takes arbitrary non-intersecting curves and evolves them into convex ones. Then, it evolves convex curves into circular curves which converge to a point [39].

In many contexts it is unusual to work with an explicit representation of the curve. When $\mathcal{C}$ is explicit, it is not easy to deal with topological changes like merge and split, and a re-parametrization of the curve may be required. The Osher-Setian [22] level set method fixes this by representing the curve implicitly as a level set of an embedding function. Let $u : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$ be an implicit representation of $\mathcal{C}$ such that $\mathcal{C}(t) = \{(x,y); u(t,(x,y)) = 0\}$. If the curve evolution has the form $\mathcal{C}_t = \mathcal{F} \cdot \mathcal{N}$, the evolution of any function $u(x,y)$ which embeds the curve as one of its level sets is $\frac{\partial u}{\partial t} = \mathcal{F} \cdot |\nabla u|$ [22], [39].

In the level-set framework, the previous PDEs for curve evolution are

$$\frac{\partial u}{\partial t} = \pm|\nabla u| \tag{1}$$

when $\mathcal{F} = \pm 1$ and

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) \cdot |\nabla u| \tag{2}$$

when $\mathcal{F} = \mathcal{K}$, since the divergence of the normalized gradient gives the curvature of the implicit curve at each point.

Most existing results for curve evolution also apply to the case of surface evolution. The explicit representation of an evolving surface is a map $\mathcal{S} : \mathbb{R}^+ \times [0,1]^2 \to \mathbb{R}^3$. The evolution rule of a surface has the form $\mathcal{S}_t = \mathcal{F}\mathcal{N}$, where $\mathcal{F}$ is a scalar field and $\mathcal{N}$ is the normal to the surface at each point. The previous curve evolution approaches have equivalent versions for surfaces when $\mathcal{F} = \pm 1$ or $\mathcal{F} = \mathcal{H}$, where $\mathcal{H}$ is the mean curvature of the surface. The extension of the level-set framework to the implicit surface evolution is straightforward. Consider the scalar field $u : \mathbb{R}^+ \times \mathbb{R}^3 \to \mathbb{R}$. A surface $\mathcal{S}$ is implicitly defined as a level set of u. The expressions for the implicit surface evolution coincide with equation (1) when $\mathcal{F} = \pm 1$ and with equation (2) when $\mathcal{F} = \mathcal{H}$. Equation (2) defined for the general n-dimensional case is known as the *mean curvature motion*.

2.2 Morphological operators

Monotone contrast-invariant and translation-invariant operators are called *morphological operators*. The most common ones are the dilation and the erosion operators. A dilation D_h with radius h of function u is defined as

$$D_h u(\mathbf{x}) = \sup_{\mathbf{y} \in hB(\mathbf{0},1)} u(\mathbf{x}+\mathbf{y}), \tag{3}$$

while the erosion has a similar form

$$E_h u(\mathbf{x}) = \inf_{\mathbf{y} \in hB(\mathbf{0},1)} u(\mathbf{x}+\mathbf{y}). \tag{4}$$

In both definitions, $B(\mathbf{0},1)$ is the ball of radius 1 centered at $\mathbf{0}$ and the term hB is the set B scaled by h, i.e., $hB = \{hx : x \in B\}$.

An interesting result in mathematical morphology is that every morphological operator T admits a *sup-inf* representation of the form

$$(T_h u)(\mathbf{x}) = \sup_{B \in \mathcal{B}} \inf_{\mathbf{y} \in \mathbf{x}+hB} u(\mathbf{y}) \tag{5}$$

or a dual *inf-sup* representation

$$(T_h u)(\mathbf{x}) = \inf_{B \in \mathcal{B}} \sup_{\mathbf{y} \in \mathbf{x}+hB} u(\mathbf{y}). \tag{6}$$

In both cases, $\mathcal{B}$ is a set of structuring elements that uniquely defines the operator, and h is the size of the operator.

For example, choosing a proper $\mathcal{B}$ one may express dilations and erosions in a *sup-inf* or *inf-sup* form. The dilation with radius h admits an *inf-sup* form when $\mathcal{B}$ is made of the single structuring element, $\mathcal{B} = \{B(\mathbf{0},1)\}$. Similarly, the erosion has an *sup-inf* form using the same base.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

2.3 From PDEs to morphological operators

Some morphological operators can be expressed as PDEs. The key idea to find these connections is to study the behavior of the successive application of a morphological operator with a very small radius. In this section we review some of these relations.

The dilation D_h verifies that [37]

$$\lim_{h\to 0^+}\frac{D_h u - u}{h} = |\nabla u|. \tag{7}$$

This means that the successive application of D_h with very small radius, $\lim_{m\to\infty}(D_{t/m})^m u_0$, is equivalent to the solution of

$$\frac{\partial u}{\partial t} = |\nabla u| \tag{8}$$

with initial value $u(0,\mathbf{x}) = u_0(\mathbf{x})$. We say that the dilation has an *infinitesimal behavior* equivalent to the PDE (8).

The erosion presents a similar property, since [37]

$$\lim_{h\to 0^+}\frac{E_h u - u}{h} = -|\nabla u|, \tag{9}$$

and therefore the erosion has an infinitesimal behavior equivalent to the PDE

$$\frac{\partial u}{\partial t} = -|\nabla u| \tag{10}$$

Thanks to this behavior, we can approximate the level-set evolution PDEs (1) using the successive application of the morphological operators D_h and E_h.

3 The Curvature Morphological Operator

In this section we introduce a new curvature morphological operator that can be used to evolve curves embedded in spaces of any dimension.

3.1 The 2D curvature morphological operator

Let SI_h and IS_h be respectively the *sup-inf* and *inf-sup* morphological operators given by the base $\mathcal{B}^2 = \{[-1,1]_\theta \subset \mathbb{R}^2 : \theta \in [0,\pi)\}$ made of all segments of length 2 centered at the origin. In their pioneering work, Catté, Dibos and Koepfler proved that the successive application of the *mean operator*, $F_{\sqrt{h}}$, for a very small h is equivalent to the curvature flow of the PDE (2) [41], where

$$(F_h u)(\mathbf{x}) = \frac{(SI_{2h}\,u)(\mathbf{x}) + (IS_{2h}\,u)(\mathbf{x})}{2}. \tag{11}$$

Unfortunately, operator F_h is not contrast-invariant and hence it is not a morphological operator. We avoid this problem using operator composition.

Lemma 3.1. *Let T_h^1 and T_h^2 be two morphological operators, we have, for a small h, that*

$$T_{h/2}^2 \circ T_{h/2}^1 u \approx \frac{T_h^2 u + T_h^1 u}{2}. \tag{12}$$

Proof: Let L_h^1 and L_h^2 be the corresponding infinitesimal operators of morphological operators T_h^1 and T_h^2. We can write a first order approximation to the operator composition $T_{h/2}^2 \circ T_{h/2}^1$ as

$$\begin{aligned} T_{h/2}^2 \circ T_{h/2}^1 u &\approx T_{h/2}^2\left(u + \frac{h}{2}L_h^1(u)\right) \\ &\approx u + \frac{h}{2}L_h^1(u) + \frac{h}{2}L_h^2(u) + \\ &\quad + \left(\frac{h}{2}\right)^2 L_h^2 L_h^1(u). \end{aligned}$$

The last term depends on the squared value of h, and it can be dismissed for a small h,

$$T_{h/2}^2\circ T_{h/2}^1 u \approx \frac{u + hL_h^1(u)}{2} + \frac{u + hL_h^2(u)}{2} \approx \frac{1}{2}(T_h^2 u + T_h^1 u).$$

□

The approximation in (12) is accurate for small values of h. In this case we can replace $\frac{1}{2}(T_h^2 u + T_h^1 u)$ by the composition $T_{h/2}^2 \circ T_{h/2}^1 u$. Then, the non-morphological operator $F_{\sqrt{h}}$ can be approximated by the composition $SI_{\sqrt{h}} \circ IS_{\sqrt{h}}$.

Definition 3.2. *The curvature morphological operator is defined as $SI_{\sqrt{h}} \circ IS_{\sqrt{h}}$.*

3.2 The d-dimensional curvature morphological operator

Here we generalize the curvature morphological operator to hyper-surfaces of any dimension. In this case the embedding function is defined as $u : \mathbb{R}^d \to \mathbb{R}$, and its level sets are $(d-1)$-hypersurfaces. The d-dimensional morphological operators SI_h and IS_h are, respectively, the *sup-inf* and *inf-sup* operators with the new base $\mathcal{B}^d$, made up of all hyper-disks of radius 1 centered at the origin, $\mathcal{B}^d = \{K_\mathbf{n} : \mathbf{n} \in S^{d-1}\}$, where $S^{d-1} = \{\mathbf{n} \in \mathbb{R}^d : \|\mathbf{n}\| = 1\}$ and $K_\mathbf{n} = \{\mathbf{v} \in \mathbb{R}^d : \|\mathbf{v}\| \le 1, \mathbf{v}^T\mathbf{n} = 0\}$.

The effect of the mean operator $(SI_h + IS_h)/2$ over the hyper-surfaces of u is explained by the following Theorem,

Theorem 3.3. *The d-dimensional operator F_h has an infinitesimal behavior given by*

$$\begin{aligned} \lim_{h\to 0^+}\frac{(F_{\sqrt{h}}u) - u}{h} &= (\min(\kappa_1,\ldots,\kappa_{d-1},0) \\ &\quad + \max(\kappa_1,\ldots,\kappa_{d-1},0)) \cdot |\nabla u|, \end{aligned}$$

where $\kappa_i, 1 \le i \le d-1$ are the principal curvatures of the hyper-surface implicitly defined by u at each point.

Proof: The projection matrix associated to a vector $\mathbf{n} \in S^{d-1}$ is $P_\mathbf{n} = Id - \mathbf{n}\mathbf{n}^T$, where Id is the identity $d \times d$ matrix. $P_\mathbf{n}$ projects any vector $\mathbf{w}$ to a hyperplane orthogonal to $\mathbf{n}$. We can easily deduce that $K_\mathbf{n} = \{P_\mathbf{n}\mathbf{w} : \mathbf{w} \in S^{d-1}\}$.

To study the effect of the operators, we expand u using Taylor series up to second order:

$$u(\mathbf{x} + h\mathbf{w}) = u(\mathbf{x}) + h\nabla u(\mathbf{x})^T\mathbf{w} + \frac{h^2}{2}\mathbf{w}^T D^2(\mathbf{x})\mathbf{w} + o(h^3),$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

where $D^2(\mathbf{x})$ is the Hessian matrix of u at $\mathbf{x}$. If we want to restrict the Taylor expansion to the orthogonal plane to a given vector $\mathbf{n} \in S^{d-1}$ we can use the projection matrix $P_\mathbf{n}$. That is,

$$u(\mathbf{x}+hP_\mathbf{n}\mathbf{w}) = u(\mathbf{x})+h\nabla u(\mathbf{x})^T P_\mathbf{n}\mathbf{w}+\frac{h^2}{2}\mathbf{w}^T P_\mathbf{n} D^2(\mathbf{x})P_\mathbf{n}\mathbf{w}+o(h^3).$$

A case of special interest is $\mathbf{n}_g = \frac{\nabla u(\mathbf{x})}{\|\nabla u(\mathbf{n})\|}$, which corresponds to the analysis of the geometric properties of the level set surface at $\mathbf{x}$. In this case, the above Taylor expansion can be expressed as

$$u(\mathbf{x}+hP_{\mathbf{n}_g}\mathbf{w}) = u(\mathbf{x})+\frac{h^2}{2}\mathbf{w}^T P_{\mathbf{n}_g} D^2(\mathbf{x})P_{\mathbf{n}_g}\mathbf{w}+o(h^3).$$

Note that the first order term cancels since $P_{\mathbf{n}_g}\mathbf{w}$ is orthogonal to $\nabla u(\mathbf{x})$. Matrix $M = P_{\mathbf{n}_g} D^2(\mathbf{x})P_{\mathbf{n}_g}$ is a $d \times d$ matrix with d eigenvalues. One of them is $\lambda_0 = 0$ which corresponds to the eigenvector $\mathbf{n}_g$. The $d-1$ eigenvectors λ_i, $1 \le i \le d-1$ are closely related to the principal curvatures κ_i of the implicit hyper-surface at $\mathbf{x}$:

$$\lambda_i = \kappa_i \cdot |\nabla u(\mathbf{x})|. \tag{13}$$

See the Chapter 11 of [40] for details. A straightforward computation leads to

$$\begin{aligned}\sup_{\mathbf{y}\in\mathbf{x}+hK_{\mathbf{n}_g}} u(\mathbf{y}) &= \sup_{\mathbf{w}\in S^{d-1}} u(\mathbf{x}+hP_{\mathbf{n}_g}\mathbf{w}) \qquad (14)\\ &= u(\mathbf{x})+\frac{h^2}{2}\max(\lambda_1,\ldots,\lambda_{d-1},0)+o(h^3),\end{aligned}$$

$$\begin{aligned}\inf_{\mathbf{y}\in\mathbf{x}+hK_{\mathbf{n}_g}} u(\mathbf{y}) &= \inf_{\mathbf{w}\in S^{d-1}} u(\mathbf{x}+hP_{\mathbf{n}_g}\mathbf{w}) \qquad (15)\\ &= u(\mathbf{x})+\frac{h^2}{2}\min(\lambda_1,\ldots,\lambda_{d-1},0)+o(h^3).\end{aligned}$$

We will prove that the supreme of the infimum for the SI_h operator is attained at $\mathbf{n}_g$, so that

$$SI_h u(\mathbf{x}) = \sup_{\mathbf{n}\in S^{(d-1)}} \inf_{\mathbf{y}\in hK_\mathbf{n}} u(\mathbf{y}) = \inf_{\mathbf{y}\in\mathbf{x}+hK_{\mathbf{n}_g}} u(\mathbf{y}).$$

Let us denote by $\mathbf{n}_h \in S^{d-1}$ an orthogonal direction where

$$\sup_{\mathbf{n}\in S^{(d-1)}} \inf_{\mathbf{y}\in hK_\mathbf{n}} u(\mathbf{y}) = \inf_{\mathbf{y}\in\mathbf{x}+hK_{\mathbf{n}_h}} u(\mathbf{y}). \tag{16}$$

Note that since S^{n-1} is a compact set and $n \to \inf_{\mathbf{y}\in hK_\mathbf{n}} u(\mathbf{y})$ is a continuous function, then $\mathbf{n}_h$ always exists. We will show that

$$\lim_{h\to 0} \mathbf{n}_h = \mathbf{n}_g. \tag{17}$$

To prove the equality we assume the opposite, that is, there exists $\epsilon > 0$ such that for each $m \in \mathbb{N}$ there exists $\mathbf{n}_{h_m} \in S^{d-1}$ with $\|\mathbf{n}_{h_m} - \mathbf{n}_g\| > \epsilon$ and $|h_m| < \frac{1}{m}$. We will show that if h_m is small enough, then

$$\inf_{\mathbf{y}\in\mathbf{x}+hK_{\mathbf{n}_{h_m}}} u(\mathbf{y}) < \inf_{\mathbf{y}\in\mathbf{x}+hK_{\mathbf{n}_g}} u(\mathbf{y}) \tag{18}$$

which is in contradiction with the definition of $\mathbf{n}_h$ in (16). We observe that the above inequality is equivalent to

$$\begin{aligned}&\inf_{w\in S^{n-1}}\left(\nabla u(\mathbf{x})^T P_{\mathbf{n}_{h_m}}\mathbf{w}+\frac{h_m}{2}\mathbf{w}^T P_{\mathbf{n}_{h_m}} D^2(\mathbf{x})P_{\mathbf{n}_{h_m}}\mathbf{w}+o(h_m^2)\right)\\ &\quad< \inf_{\mathbf{w}\in S^{n-1}}\left(\frac{h_m}{2}\mathbf{w}^T P_{\mathbf{n}_g} D^2(\mathbf{x})P_{\mathbf{n}_g}\mathbf{w}+o(h_m^2)\right).\end{aligned}$$

When h_m goes to 0, the right part of the above inequality goes to 0 too. However, in the left hand side, we observe that if we choose $\mathbf{w} = \mathbf{n}_g$ then $\nabla u(\mathbf{x})^T P_{\mathbf{n}_{h_\infty}}\mathbf{n}_g = \epsilon' > 0$ since $\|\mathbf{n}_{h_m} - \mathbf{n}_g\| > \epsilon$. Therefore, the left hand side holds that

$$\begin{aligned}&\inf_{w\in S^{n-1}}\left(\nabla u(\mathbf{x})^T P_{\mathbf{n}_{h_m}}\mathbf{w}+\frac{h_m}{2}\mathbf{w}^T P_{\mathbf{n}_{h_m}} D^2(\mathbf{x})P_{\mathbf{n}_{h_m}}\mathbf{w}+o(h_m^2)\right)\\ &\quad\le -\epsilon'+\frac{h_m}{2}\mathbf{n}_g^T P_{\mathbf{n}_{h_m}} D^2(\mathbf{x})P_{\mathbf{n}_{h_m}}\mathbf{n}_g+o(h_m^2).\end{aligned}$$

If h_m is small enough, the right part of the above inequality is strictly lower than 0, and (18) is satisfied. This is in contradiction with the definition of $\mathbf{n}_h$, and therefore (17) is true.

We can use the same argument for the operator IS_h: when h goes to 0, the infimum is attained in the plane orthogonal to the gradient. Therefore, when h is small, the operators SI_h and IS_h behave as given in the expressions (14) and (15). Using these expressions, the mean operator F_h

$$F_h u(\mathbf{x}) = \frac{(SI_{2h}\,u)(\mathbf{x})+(IS_{2h}\,u)(\mathbf{x})}{2}$$

can be written as

$$\begin{aligned}F_h u(\mathbf{x}) &= u(\mathbf{x})+h^2(\min(\lambda_1,\ldots,\lambda_{d-1},0)\\ &\quad+\max(\lambda_1,\ldots,\lambda_{d-1},0))+o(h^3).\end{aligned}$$

Reorganizing terms and substituting the eigenvalues of M according to (13), we obtain that

$$\begin{aligned}\lim_{h\to 0^+}\frac{F_{\sqrt{h}}u(\mathbf{x})-u(\mathbf{x})}{h} &= (\min(\kappa_1,\ldots,\kappa_{d-1},0)\\ &\quad \max(\kappa_1,\ldots,\kappa_{d-1},0))\cdot|\nabla u|.\end{aligned}$$

□

Corollary 3.4. *The 2D operator F_h has an infinitesimal behavior given by*

$$\lim_{h\to 0^+}\frac{(F_{\sqrt{h}}u)-u}{h} = \kappa\cdot|\nabla u| = div\left(\frac{\nabla u}{|\nabla u|}\right)\cdot|\nabla u|. \tag{19}$$

Proof: Trivial from Theorem 3.3. □

This is the result that Catté, Dibos and Koepfler proved in [41]. Here we have shown that it is a special case of the more general Theorem 3.3. In consequence, the successive application of the curvature morphological operator, $SI_{\sqrt{h}} \circ IS_{\sqrt{h}}$, with base $\mathcal{B}^2$ is equivalent to the solution of PDE (2).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

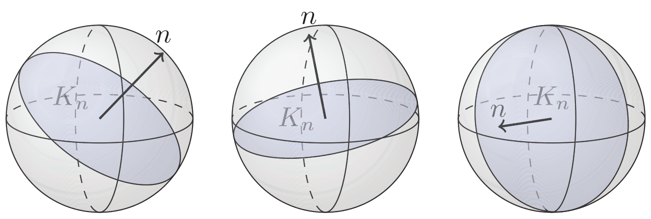

Fig. 1. The base $\mathcal{B}^3$ for the 3D SI_h and IS_h operators is made up of all disks with radius 1 centered at the origin. The figure depicts three of those disks marked in blue. Each disk K_n is determined by its normal $\mathbf{n}$.

3.3 The 3D curvature morphological operator

Although we extended in section 3.2 the definition of the curvature morphological operator to the n-dimensional case, practical applications will typically use the 2D and 3D versions to process respectively images and image stacks. In the experiments section we study these two cases. So, for completeness, here we consider the 3D curvature morphological operator and compare it with the well-known mean curvature motion PDE.

Operators SI_h and IS_h may also be expressed in the 3D case. We define the new base as the set of all disks of radius 1 centered at the origin $\mathcal{B}^3 = \{K_\mathbf{n} : \mathbf{n} \in S^2\}$, where $S^2 = \{\mathbf{n} \in \mathbb{R}^3 : \|\mathbf{n}\| = 1\}$ is the 2-sphere of radius 1 and $K_\mathbf{n} = \{\mathbf{v} \in \mathbb{R}^3 : \|\mathbf{v}\| \leq 1, \mathbf{v}^T\mathbf{n} = 0\}$ is the disk of radius 1 centered at the origin and orthogonal to $\mathbf{n}$. Figure 1 shows some elements of $\mathcal{B}^3$. Now we can state the following

Corollary 3.5. *The 3D operator F_h has the infinitesimal behavior*

$$\lim_{h\to 0^+} \frac{(F_{\sqrt{h}}u) - u}{h} = (\min(\kappa_1, \kappa_2, 0) + \max(\kappa_1, \kappa_2, 0)) \cdot |\nabla u|,$$

where κ_1 and κ_2 are the principal curvatures of the surface implicitly defined by u at each point.

Proof: Trivial from Theorem 3.3. □

Let us now compare the infinitesimal behavior of the 3D operator F_h with the mean curvature motion PDE (2) in 3D. Recall that the term $\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$ is equivalent to the mean curvature $\mathcal{H} = \kappa_1 + \kappa_2$ of the implicit surface. Therefore, the mean curvature motion may be rewritten as

$$\frac{\partial u}{\partial t} = (\kappa_1 + \kappa_2) \cdot |\nabla u|. \tag{20}$$

Corollary 3.5 relates the evolution provided by $SI_{\sqrt{h}} \circ IS_{\sqrt{h}}$ with the evolution of PDE

$$\frac{\partial u}{\partial t} = (\min(\kappa_1, \kappa_2, 0) + \max(\kappa_1, \kappa_2, 0)) \cdot |\nabla u|. \tag{21}$$

Comparing PDEs (20) and (21) we can immediately conclude that the evolution provided by the curvature morphological operator is not strictly equivalent to the mean curvature flow. We can distinguish two cases. First, if both principal curvatures have the same sign (e.g., a sphere) and assuming that $|\kappa_2| \geq |\kappa_1|$, (21) becomes

$$\frac{\partial u}{\partial t} = \kappa_2 \cdot |\nabla u|.$$

This is equivalent to the mean curvature motion if $\kappa_1 = 0$. As $|\kappa_1|$ gets larger, the difference between both flows becomes more noticeable. However, the sign of the flow is the same in both equations. In the second case, the principal curvatures have different sign (e.g., a catenoid). Then, (21) becomes

$$\frac{\partial u}{\partial t} = (\kappa_1 + \kappa_2) \cdot |\nabla u|,$$

which coincides with the mean curvature motion PDE.

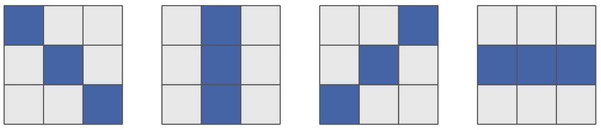

Fig. 2. The base $\mathcal{P}$ for the 2D discrete $SI_d \circ IS_d$ operator.

3.4 The discrete curvature morphological operator

The embedding function u has to be discretized to be used in practical applications. Usually, the discretization of u consists of an orthogonal grid of cells with constant values within each cell. These cells are often called *pixels* when u is two-dimensional and *voxels* when it is three-dimensional. When u is discrete, i.e. $u : \mathbb{Z}^d \to \mathbb{R}$, expression $u(\mathbf{x})$ refers to the value of the cell at position $\mathbf{x}$.

Working with discrete functions, the curvature morphological operator must be discretized accordingly, which is equivalent to discretize its base $\mathcal{B}$. For the 2D case, we choose the four discrete segments of three pixels of length in all possible orientations,

$$\mathcal{P} = \left\{ \begin{array}{c} \{(0,0),(1,0),(-1,0)\}, \\ \{(0,0),(0,1),(0,-1)\}, \\ \{(0,0),(1,1),(-1,-1)\}, \\ \{(0,0),(1,-1),(-1,1)\} \end{array} \right\}. \tag{22}$$

See Figure 2 for a graphical representation. For the 3D operator, we take the nine discrete planes of 3×3 voxels in all possible orientations (see Figure 3).

We use expression $SI_d \circ IS_d$ to denote the *discrete curvature morphological operator*. This notation is the same for all the dimensions. Note that here d stands for *discrete*, and it is not the scale of the operator. In the discrete version we do not need to explicitly indicate the scale as we will always use the smallest one.

3.5 How does the $SI_d \circ IS_d$ operator work?

One of the overall effects of the $SI_d \circ IS_d$ operator is the smoothing of the implicit hyper-surfaces of u. Here we present an intuitive explanation of how the smoothing is achieved with the 2D $SI_d \circ IS_d$ operator.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

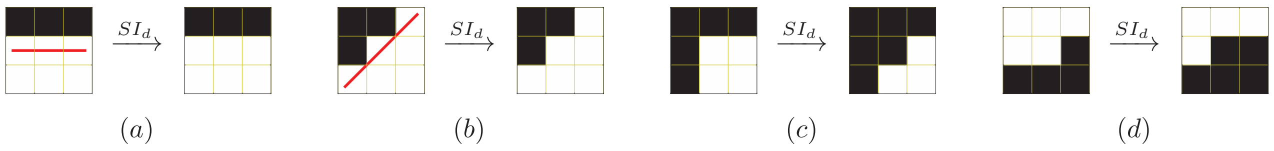

Fig. 4. Some examples of the effect of the SI_d operator on individual pixels of binary images. In those cases where a straight line is found (marked in red), the central pixel remains active ((a) and (b)). When the central pixel does not belong to a straight line of active pixels, it is made inactive ((c) and (d)). For exemplification purposes, we assume the pixels on the borders are not affected by the operator.

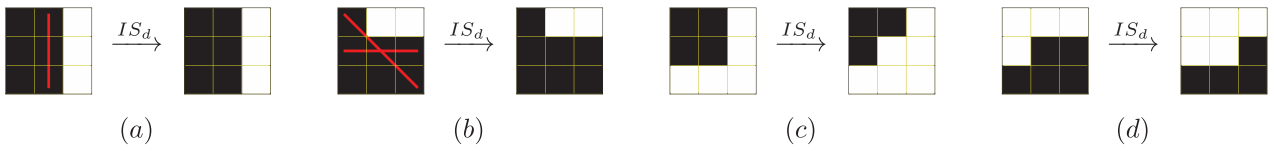

Fig. 5. Idem for the IS_d operator.

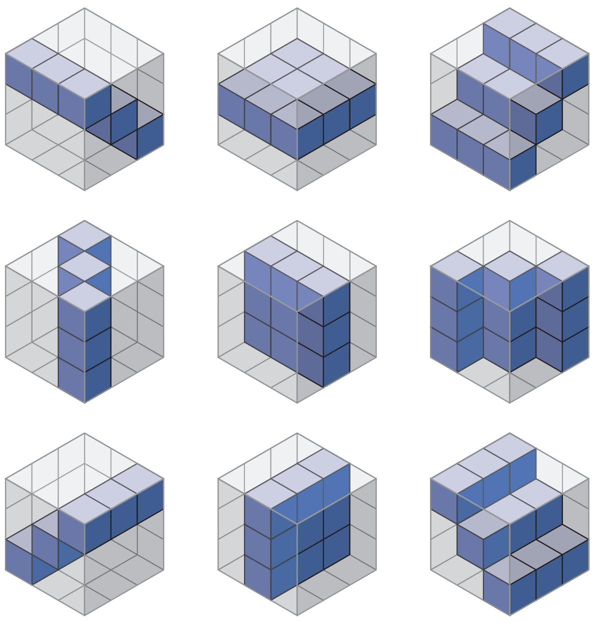

Fig. 3. The base $\mathcal{P}$ for the 3D discrete $SI_d \circ IS_d$ operator.

In a discrete binary function u, both SI_d and IS_d perform the same operation, but SI_d works only on white (or *active*) pixels and IS_d only on black (or *inactive*) pixels. It is easy to see that SI_d does not affect inactive pixels. Suppose $u(\mathbf{x}_0)$ is an inactive pixel, i.e., $u(\mathbf{x}_0) = 0$. Then, $\inf_{\mathbf{y}\in\mathbf{x}_0+P} u(\mathbf{y})$ will be 0 for every segment P in $\mathcal{P}$, and therefore $(SI_d\, u)(\mathbf{x}_0) = 0$. Following a similar reasoning, we can see that IS_d does not affect active pixels.

For every active pixel $\mathbf{x}_1$ in a binary image, the SI_d operator looks for small (3 pixels long) straight lines of active pixels which contain $\mathbf{x}_1$. This search is done in the four possible orientations corresponding to the four segments in $\mathcal{P}$. If no straight line exists, the pixel is made inactive (see Figure 4). Sharp edges (Fig. 4c and 4d) are detected as those pixels which are not part of a straight line and removed. The active pixels in smooth edges (Fig. 4a and 4b) remain unchanged.

For inactive pixels, the IS_d operator carries out a similar procedure (see Figure 5).

The composition $SI_d \circ IS_d$ first removes the sharp inactive pixels with IS_d, and then repeats the procedure for the active ones with SI_d. The result is a global smoothing of u, as can be seen in the first row of Figure 6.

4 Morphological Snakes

Now we have a set of morphological operators — dilation, erosion and the new curvature flow operator $SI_d \circ IS_d$— which have an infinitesimal behavior like PDEs (1) and (2) respectively. These two PDEs are fundamental in many practical applications, since they are part of many contour evolution rules. Thus, given a PDE of contour evolution which includes terms (1) and (2), we may compose their corresponding morphological operators to approximate the solution. In other words, now we can use mathematical morphology to evolve contours. In the following sections, we apply this idea to morphologically solve some well-known contour evolution PDEs: the GAC [20] and the ACWE models [10].

4.1 Morphological GAC

In the GAC framework, an energy functional, which depends on the contents of an image I, is assigned to a curve,

$$\begin{aligned} E(\mathcal{C}) &= \int_0^{\text{length}(\mathcal{C})} g(I)(\mathcal{C}(s))ds \qquad (23)\\ &= \int_0^1 g(I)(\mathcal{C}(p)) \cdot |\mathcal{C}_p|dp, \end{aligned}$$

or surface,

$$E(\mathcal{S}) = \iint g(I)(\mathcal{S}(a))da,$$

where $ds = |\mathcal{C}_p|dp$ is the Euclidean arc-length parametrization of the curve, da is the Euclidean element

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

of area, and $g(I) : \mathbb{R}^d \to \mathbb{R}^+,\ \mathbf{x} \to g(I)(\mathbf{x})$ allows us to select which regions of the image we are interested in. Typically, $g(I)$ could be

$$g(I) = \frac{1}{\sqrt{1 + \alpha|\nabla G_\sigma * I|}}, \tag{24}$$

which is low in the edges of the image, or $g(I) = |G_\sigma * I|$, which attains its minima in the center of the image dark lines. $G_\sigma *$ is a gaussian filter with standard deviation σ. The geodesic active contours model does not depend on the parametrization of the curve or surface, and the minimum energy hyper-surfaces

$$\begin{aligned} \mathcal{C}^* &= \arg\min_{\mathcal{C}} E(\mathcal{C}) \\ \mathcal{S}^* &= \arg\min_{\mathcal{S}} E(\mathcal{S}) \end{aligned}$$

correspond to the geodesics of a Riemannian space whose metric is defined by $g(I)$. These hyper-surfaces tend to be smooth and to pass through low values of the function $g(I)$.

The minimization of the energy functionals is done in a steepest-descent way. The Euler-Lagrange equation of the functional gives the direction of the descent. The local minima are reached at the steady states of the differential equation

$$\mathcal{C}_t = (g(I) \cdot \mathcal{K} - \nabla g(I) \cdot \mathcal{N})\mathcal{N}$$

for curves and

$$\mathcal{S}_t = (g(I) \cdot \mathcal{H} - \nabla g(I) \cdot \mathcal{N})\mathcal{N}$$

for surfaces, with the given initial values $\mathcal{C}(0) = \mathcal{C}_0$ and $\mathcal{S}(0) = \mathcal{S}_0$.

Sometimes, the attraction force is not strong enough to move the hyper-surface (because the field $\nabla g(I)$ is too small or because this field and the normal are orthogonal). Hence, portions of the hyper-surface usually get stuck in these non-informative areas. To overcome the problem, a common solution is the introduction of the so-called *balloon force* [43]. The evolution with the auxiliary balloon force is

$$\mathcal{C}_t = (g(I)\mathcal{K} + g(I)\nu - \nabla g(I) \cdot \mathcal{N})\mathcal{N} \tag{25}$$

or

$$\mathcal{S}_t = (g(I)\mathcal{H} + g(I)\nu - \nabla g(I) \cdot \mathcal{N})\mathcal{N}$$

where $\nu \in \mathbb{R}$ is the balloon force parameter.

These expressions can be rewritten in terms of a level set implementation as

$$\frac{\partial u}{\partial t} = g(I)|\nabla u| \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + g(I)|\nabla u|\nu + \nabla g(I)\nabla u. \tag{26}$$

The flow given by this expression has three components. The smoothing force, which tends to smooth the hyper-surface at high curvature segments; the balloon force, which *inflates* or *deflates* the hyper-surface in areas of little information; and the image attraction force, which is responsible for bringing the hyper-surface to the interesting regions of the image.

Inspired by the similarities between the GAC PDE (26) and PDEs (7), (9) and (19) describing the infinitesimal behavior of morphological operators, we propose a fast and stable curve evolution approach based on mathematical morphology. The new evolution will use a combination of binary morphological operators whose infinitesimal behavior is similar to the flow expressed by the equation (26). Binary operators require a binary embedding function u. Therefore, the hyper-surface is given as the level set $\frac{1}{2}$ of a binary piecewise constant function $u : \mathbb{Z}^d \to \{0, 1\}$. We take $u(\mathbf{x}) = 1$ for every point $\mathbf{x}$ inside the hyper-surface, and $u(\mathbf{x}) = 0$ for every point $\mathbf{x}$ outside. The morphological operators will act on u and, hence, they will implicitly evolve the hyper-surface.

4.1.1 Balloon force

We focus on the balloon type operator term of equation (26):

$$\frac{\partial u}{\partial t} = g(I) \cdot \nu \cdot |\nabla u|. \tag{27}$$

The factor $g(I)$ controls the strength of the balloon force in different fragments of the hyper-surface: when $g(I)$ is high, the corresponding fragment is located far from a target region, and the balloon force must be strong; on the other hand, when $g(I)$ becomes lower, the hyper-surface is approaching its objective, and hence the balloon force becomes unnecessary. The effect of the $g(I)$ factor in (27) can be discretized with a single threshold θ: when $g(I)$ is greater than θ, the corresponding point is updated according to the balloon force, and left unchanged otherwise. Depending on the sign of ν, the remaining factors $\nu \cdot |\nabla u|$ lead to the dilation and the erosion PDEs (1). Given the hyper-surface status at iteration n, $u^n : \mathbb{Z}^d \to \{0, 1\}$, the balloon force PDE (27) applied over u^n can be approximated using the following morphological approach:

$$u^{n+1}(\mathbf{x}) = \begin{cases} (D_d u^n)(\mathbf{x}) & \text{if } g(I)(\mathbf{x}) > \theta \text{ and } \nu > 0 \\ (E_d u^n)(\mathbf{x}) & \text{if } g(I)(\mathbf{x}) > \theta \text{ and } \nu < 0 \\ u^n(\mathbf{x}) & \text{otherwise} \end{cases}.$$

D_d and E_d are the discrete versions of dilation and erosion.

4.1.2 Smoothing force

The smoothing force term of (26),

$$\frac{\partial u}{\partial t} = g(I) \cdot |\nabla u| \cdot \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right), \tag{28}$$

is a weighted version of the mean curvature motion PDE (2). As in the previous case, the $g(I)$ factor acts like a weight which controls the strength of the smoothing operation at every point. We could discretize it again by means of a threshold θ. However, in our experiments we have seen that this threshold is unnecessary. In Corollary 3.4 and Section 3.4 we prove that the discrete

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

morphological curvature operator, $SI_d \circ IS_d$, has an infinitesimal behavior equivalent to the mean curvature motion PDE (2). Thus, the morphological equivalent of PDE (28) is

$$u^{n+1}(\mathbf{x}) = \left((\underset{d}{SI} \circ \underset{d}{IS})^{\mu} u^n \right)(\mathbf{x})$$

The number of successive applications of the smoothing operator controls the strength of the smoothing step. This number is indicated by parameter $\mu \in \mathbb{N}$.

4.1.3 Solving the complete GAC PDE

As we stated above, the active contour equation (26) is made up of three different components: a smoothing force, a balloon force and an attraction force. We have seen how two of these components may be solved with morphological operators. The third component, i.e., the attraction force, has an immediate discrete version as we will see below.

In the PDE, the combination of the three components is performed through their addition. Our morphological solution will combine them by composition: in each iteration, we will apply the morphological balloon (27), the morphological smoothing (28) and the discretized attraction force over the embedding level set function u. Given the snake status at iteration n, u^n, we get u^{n+1} using the following steps:

$$\begin{aligned}
u^{n+\frac{1}{3}}(\mathbf{x}) &= \begin{cases} (D_d u^n)(\mathbf{x}) & \text{if } g(I)(\mathbf{x}) > \theta \\ & \quad \text{and } \nu > 0 \\ (E_d u^n)(\mathbf{x}) & \text{if } g(I)(\mathbf{x}) > \theta \\ & \quad \text{and } \nu < 0 \\ u^n(\mathbf{x}) & \text{otherwise} \end{cases} \\
u^{n+\frac{2}{3}}(\mathbf{x}) &= \begin{cases} 1 & \text{if } \nabla u^{n+\frac{1}{3}} \nabla g(I)(\mathbf{x}) > 0 \\ 0 & \text{if } \nabla u^{n+\frac{1}{3}} \nabla g(I)(\mathbf{x}) < 0 \\ u^{n+\frac{1}{3}} & \text{if } \nabla u^{n+\frac{1}{3}} \nabla g(I)(\mathbf{x}) = 0 \end{cases} \qquad (29) \\
u^{n+1}(\mathbf{x}) &= \left((\underset{d}{SI} \circ \underset{d}{IS})^{\mu} u^{n+\frac{2}{3}} \right)(\mathbf{x})
\end{aligned}$$

which is the morphological implementation of the geodesic active contour PDE.

4.2 Morphological ACWE

Chan and Vese [10] define an energy functional for image segmentation which takes into account the content of the interior and exterior regions of the curve (or surface) in contrast to the GAC, which only take into account the places where the curve (or surface) passes. The ACWE functional of a curve $\mathcal{C}$ is

$$\begin{aligned}
F(c_1, c_2, \mathcal{C}) &= \mu \cdot \text{length}(\mathcal{C}) + \nu \cdot \text{area}(\text{inside}(\mathcal{C})) \\
&\quad + \lambda_1 \int_{\text{inside}(\mathcal{C})} \|I(\mathbf{x}) - c_1\| d\mathbf{x} \qquad (30) \\
&\quad + \lambda_2 \int_{\text{outside}(\mathcal{C})} \|I(\mathbf{x}) - c_2\| d\mathbf{x},
\end{aligned}$$

where the non-negative parameters μ, ν, λ_1 and λ_2 control the strength of each term. The three-dimensional version of this functional $F(c_1, c_2, \mathcal{S})$ is obtained by replacing the operators length by area and area by volume.

The minimization of functional

$$\min_{c_1, c_2, \mathcal{C}} F(c_1, c_2, \mathcal{C}) \text{ or } \min_{c_1, c_2, \mathcal{S}} F(c_1, c_2, \mathcal{S})$$

is slightly challenging, since it has two additional scalar variables which were not present in the geodesic active contour model. However, given a fixed contour, the values of c_1 and c_2 which minimize F are the mean of the values of I inside and outside the contour. For curves,

$$c_1(\mathcal{C}) = \frac{\int_{\text{inside}(\mathcal{C})} I(\mathbf{x}) d\mathbf{x}}{\int_{\text{inside}(\mathcal{C})} d\mathbf{x}}, \quad c_2(\mathcal{C}) = \frac{\int_{\text{outside}(\mathcal{C})} I(\mathbf{x}) d\mathbf{x}}{\int_{\text{outside}(\mathcal{C})} d\mathbf{x}}.$$

The Euler-Lagrange equation for the implicit version of functional (30) is [10]

$$\begin{aligned}
\frac{\partial u}{\partial t} &= |\nabla u| \left(\mu \operatorname{div}\left(\frac{\nabla u}{|\nabla u|} \right) - \nu \right. \qquad (31) \\
&\qquad \left. - \lambda_1 (I - c_1)^2 + \lambda_2 (I - c_2)^2 \right),
\end{aligned}$$

which is valid for a $(d-1)$-hypersurface defined as a level set of $u : \mathbb{R}^d \to \mathbb{R}$. This equation specifies how the implicit hyper-surface should evolve to minimize functional F in a steepest descent manner. It has a smoothing and a balloon term, which are treated as in the GAC model: the $SI_d \circ IS_d$ approximate the smoothing term and the erosion and dilation approximate the balloon. The *image attachment* term is new, but deriving its morphological approximation is not difficult. When $\lambda_1 |\nabla u| (I - c_1)^2 < \lambda_2 |\nabla u| (I - c_2)^2$ at $\mathbf{x}$, $\mathbf{x}$ belongs to the interior of the curve; if the inequality is reversed, $\mathbf{x}$ belongs to the exterior of the curve; otherwise, $\mathbf{x}$ remains where it was. As before, the hyper-surface must be defined as the level set $\frac{1}{2}$ of a binary embedding function $u : \mathbb{Z}^d \to \{0, 1\}$.

The morphological ACWE algorithm is given by the following three steps:

$$\begin{aligned}
u^{n+\frac{1}{3}}(\mathbf{x}) &= \begin{cases} (D_d u^n)(\mathbf{x}) & \text{if } \nu > 0 \\ (E_d u^n)(\mathbf{x}) & \text{if } \nu < 0 \\ u^n(\mathbf{x}) & \text{otherwise} \end{cases} \qquad (32) \\
u^{n+\frac{2}{3}}(\mathbf{x}) &= \begin{cases} 1 & \text{if } |\nabla u^{n+\frac{1}{3}}| (\lambda_1 (I - c_1)^2 \\ & \quad - \lambda_2 (I - c_2)^2)(\mathbf{x}) < 0 \\ 0 & \text{if } |\nabla u^{n+\frac{1}{3}}| (\lambda_1 (I - c_1)^2 \\ & \quad - \lambda_2 (I - c_2)^2)(\mathbf{x}) > 0 \\ u^{n+\frac{1}{3}} & \text{otherwise} \end{cases} \\
u^{n+1}(\mathbf{x}) &= \left((SI_d \circ IS_d)^{\mu} u^{n+\frac{2}{3}} \right)(\mathbf{x}).
\end{aligned}$$

This expression may be used in a fast, simple, stable and robust method for minimizing the ACWE functional [10].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

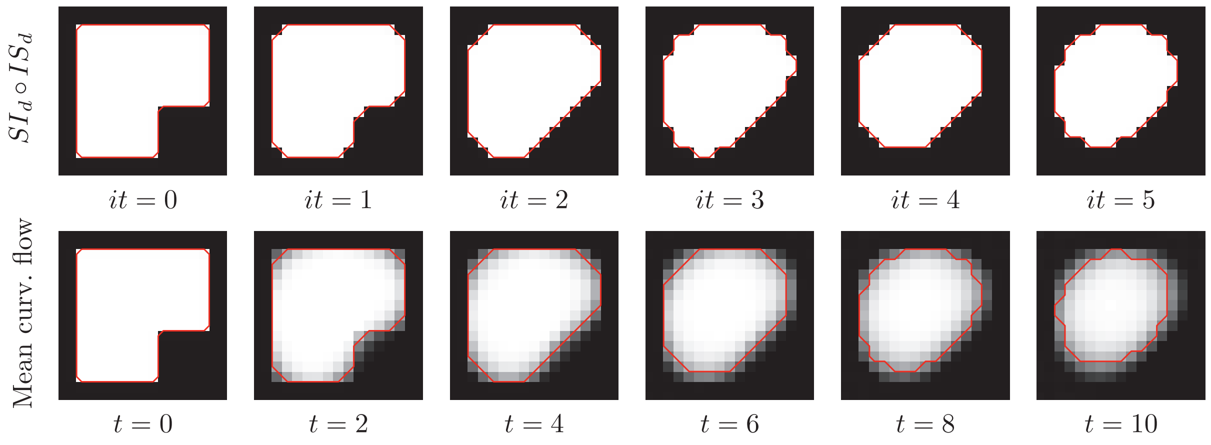

Fig. 6. Evolution of a 2D shape with the discrete curvature morphological operator $SI_d \circ IS_d$ (upper row) and the mean curvature PDE (bottom row). Background images represent the embedding function u. The red curves are the $\frac{1}{2}$ level set. Below each image we display the number of iterations (upper row) and the time parameter (lower row).

5 Implementation Details

The implementation of equations (29) and (32) is straightforward, but some details are worth mentioning. The embedding function u is stored as a d-dimensional array with a Boolean values at each cell. The dilation at each cell is implemented as the maximum of the values of u in the neighborhood of the cell. Similarly, the erosion is the minimum of the values of u in the neighborhood of the cell. The *neighborhood* is defined as the Moore neighborhood, i.e., the set of cells at a Chebyshev distance of 1. For example, in the two-dimensional case, the dilation is

$$u^{n+1}(i,j) = \max_{\Delta i, \Delta j \in \{-1,0,1\}} u^n(i+\Delta i, j+\Delta j)$$

and the erosion is

$$u^{n+1}(i,j) = \min_{\Delta i, \Delta j \in \{-1,0,1\}} u^n(i+\Delta i, j+\Delta j).$$

The gradient is the d-dimensional vector made up of all directional derivatives, $\nabla u = [u_x, u_y, \ldots,]^\top$.

Derivatives are computed using central differences. For example, in 2D, the derivatives of u with respect of x and y are computed as

$$u_x(i,j) = \frac{1}{2}\left(u(i+1,j) - u(i-1,j)\right), \quad (33)$$

$$u_y(i,j) = \frac{1}{2}\left(u(i,j+1) - u(i,j-1)\right). \quad (34)$$

Finally, the order of operator composition SI_d and IS_d in (12) could be either $SI_d \circ IS_d$ or $IS_d \circ SI_d$, since the addition is commutative. Throughout the paper we have chosen the first one. However, in practice, to balance the contribution of both operator composition choices we alternate them through iterations. The $SI_d \circ IS_d$ operator is computed in two steps: first, the IS_d step which in 2D is

$$u^{n+1}(i,j) = \min_{P \in \mathcal{P}} \max_{(\Delta i, \Delta j) \in P} u^n(i+\Delta i, j+\Delta j),$$

and then the SI_d step which in 2D is

$$u^{n+1}(i,j) = \max_{P \in \mathcal{P}} \min_{(\Delta i, \Delta j) \in P} u^n(i+\Delta i, j+\Delta j).$$

Operator $IS_d \circ SI_d$ performs these steps in reverse order. $\mathcal{P}$ is defined in (22) and shown in Figure 2. For a 3D embedding function the base $\mathcal{P}$ should be as in Figure 3. In an efficient implementation, inactive cells should be ignored in the erosion and in the SI_d operator, and active cells should be ignored in the dilation and in the SI_d operator.

This implementation is well suited for parallelization in SIMD architectures, such as GPUs. For an efficient single thread implementation, one should consider using the narrow band technique, which is not difficult to introduce. Broadly speaking, one should maintain two lists of cells —the list of outside boundary cells and the list of inside boundary cells— and apply the above implementation only over the cells in these lists. After each step, the lists should be updated accordingly.

In the experiments section we compare the performance of the morphological and a numerical solution for the GAC and ACWE algorithms. Numerical implementations of the algorithms are more complex and require a broader set of mathematical tools than than the morphological ones. Also, numerical algorithms may degenerate the level-set function, which eventually becomes too flat and no longer a signed distance function. This has to be explicitly fixed. We use the approach of [44], [45], based on solving

$$\begin{cases} \psi_\tau = \text{sign}(u(t))(1 - |\nabla \psi|) \\ \psi(0,\cdot) = u(t,\cdot), \end{cases} \quad (35)$$

where $u(t,\cdot)$ is the level set function at time t. We discretize (35) with Godunov's method [45]. The steady state of (35) is a signed distance function with the same contour as $u(t,\cdot)$. However, we do not iterate until reaching the steady state. Instead, we have found that running one iteration of the scheme in (35) after each

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

iteration of the scheme for the numerical solution suffices to prevent the degeneration of the level-set function. This is the approach used in our experiments.

As in the morphological implementation, we discretize the level set function u using a two-dimensional array. Here, each cell stores a floating point value. We use the central differences scheme for the spatial discretization of the PDEs. The grid spacing is fixed to 1, so the first derivatives are given by (33) and (34). The second derivatives are

$$\begin{aligned} u_{xx}(i,j) &= u(i+1,j)+u(i-1,j)-2u(i,j),\\ u_{yy}(i,j) &= u(i,j+1)+u(i,j-1)-2u(i,j),\\ u_{xy}(i,j) &= \frac{1}{4}(u(i+1,j+1)-u(i-1,j+1)\\ &\quad -u(i+1,j-1)+u(i-1,j-1)). \end{aligned}$$

The curvature is discretized as

$$\operatorname{div}\left(\frac{\nabla u}{\|\nabla u\|}\right)=\frac{u_{xx}u_y^2-2u_{xy}u_xu_y+u_{yy}u_x^2}{\left(u_x^2+u_y^2\right)^{\frac{3}{2}}}.$$

The other terms in PDEs (26) and (31) are straightforward to compute using the first and second order derivatives.

The discretization of the time is done with the forward (explicit) method. The time step Δt is dynamically determined so that the CFL condition is met but the evolution is not too slow.

6 Experimental Results

Besides its theoretical relevance, the morphological framework introduced in this work also presents advantages in terms of required computational resources, simplicity and stability. The aim of the experiments conducted is comparing qualitatively and quantitatively the performance of the morphological and numerical evolution algorithms.

6.1 Smoothing

First, we qualitatively compare the smoothing achieved with the discrete curvature morphological operator, $SI_d \circ IS_d$, and that obtained with the numerical solution of the mean curvature PDE (2). Figures 6 and 7 compare both methods in 2D and 3D respectively. The 2D case shown in Figure 6 starts the evolution at a 13×13 pixels square with a corner of size 5×5 pixels removed. In Figure 7, the initial shape is a $16 \times 16 \times 16$ pixels cube with the $7 \times 7 \times 7$ pixels corner removed. In the 2D (3D) experiments we can see that both the numerical and morphological approaches smooth the interface gradually towards a circular (spherical) shape. In the 2D evolution case (see Figure 6) the curve at iteration number 5 (last column) in the morphological algorithm is most similar to the numerical mean curvature result for time $t = 8$. Something similar occurs for the 3D evolution at iteration number 6 and time $t = 8$. In both cases it is not easy to decide which is the best pair of curves to compare, since we do not have an analytic relation between parameter t of the numerical mean curvature evolution and the number of iterations of the discrete morphological curvature approach. Nevertheless, the qualitative analysis of Figures 6 and 7 confirms that the morphological and numerical smoothing operators of 2D and 3D interfaces is very similar.

6.2 Contour evolution

Our next group of experiments assess the performance of the morphological approaches of contour evolution and compare their results with the numerical solution of their associated PDEs. We have implemented the GAC (26), ACWE (31), Morphological GAC (29) and Morphological ACWE (32) models in C++ (only the 2D approaches) and Python (both 2D and 3D approaches). For comparison purposes, all implementations are single-threaded, and they do not use improvement methods such as multi-scale or narrow-band solutions, although all approaches would equally benefit from them. We run the experiments on a Intel Core 2 Duo 2.4 GHz. When given, evolution times always refer to those obtained with the C++ implementation.

Determining the parameters of the morphological algorithms is not a complex task. The Morphological GAC approach of (29) has parameters θ, ν and μ. The smoothing strength μ serves as a scaling value. When we look for small elements, μ should be small (i.e., 1) and large otherwise. The parameter θ —the threshold of the balloon and the smoothing— depends on $g(I)$. A good starting point is to set θ as the 40th percentile of $g(I)$ (see Figure 8). The stopping criterion g of (24) has two parameters: α and σ. Parameter σ is assigned to match the size of the image borders. The algorithm is not very dependent on parameter α. A high value, such as 1000 or 10000 should be enough in most cases.

The parameters of the Morphological ACWE are much simpler to set up and less sensitive to perturbations than the Morphological GAC. This approach does not require a threshold θ. The balloon force is seldom necessary. It works directly on the image I. It does not use a stopping criterion g. For the experiments, we have set $\lambda_1 = \lambda_2 = 1$ and $\nu = 0$. The strength of the smoothing μ behaves as in the morphological GAC. It should be low when we look for small elements and large otherwise. Table 1 summarizes the parameter values used in our experiments.

In Figures 9 and 10 we compare the performance of the morphological and the numerical GAC in challenging conditions. In Figure 9 we run both algorithms on a noisy ultrasound image of a breast nodule. Although the final results are similar, the morphological method is almost one order of magnitude faster than its numerical counterpart (see Table 2). This experiment also confirms that the Morphological GAC works well in very noisy conditions.

The starfish image in Figure 10 has non-convex parts, elongated limbs and sharp angles at the junction of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

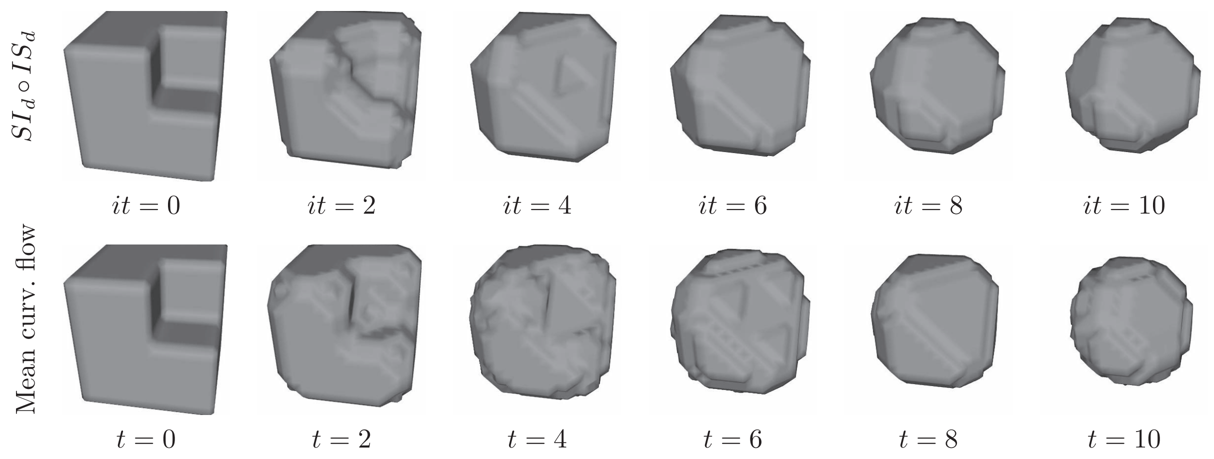

Fig. 7. Evolution of a 3D shape with the discrete curvature morphological operator $SI_d \circ IS_d$ (row 1) and the mean curvature PDE (row 2). The shapes correspond to the $\frac{1}{2}$ level set.

		Morphological GAC					GAC			
	Image size	α	σ	μ	θ	ν	α	σ	μ	ν
Breast nodule	256×256	1000	$\sqrt{30}$	1	0.31	1	1000	$\sqrt{30}$	0.1	1
Starfish	275×323	1000	2	1 & 2	0.3	-1	100000	2	0.1	-0.2

		Morphological ACWE				ACWE			
	Image size	μ	λ_1	λ_2	ν	μ	λ_1	λ_2	ν
Lakes	286×231	3	1	1	0	0.2	1	1	0
Europe	240×185	1	1	1	0	0.05	1	1	0
Dendrite	$39 \times 200 \times 200$	1	1	1	0	–	–	–	–

TABLE 1
Values for the parameters used in the experiments.

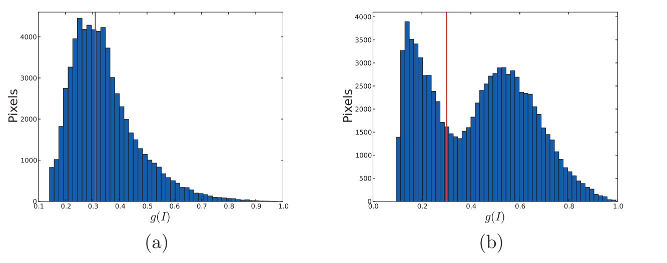

Fig. 8. Histogram of $g(I)$ for (a) the breast nodule and (b) the starfish images. The threshold θ, marked in red, is set around the $40th$ percentile in each case.

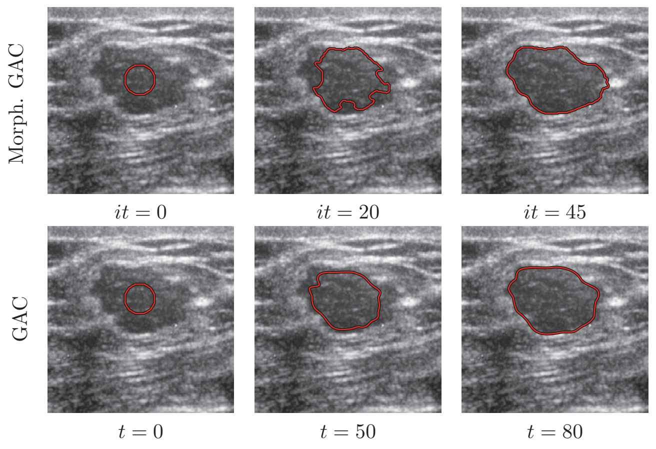

Fig. 9. Segmentation of a noisy ultrasound image. GAC initialization: $u_0 = 20 - \sqrt{(x-126)^2 + (y-100)^2}$. Morphological GAC initialization: $u_0 = T\left[20 - \sqrt{(x-126)^2 + (y-100)^2} > 0\right]$.

limbs. This is a challenging situation for the GAC (see bottom row of Figure 10), since it tends to shorten long and thin structures and to over-smooth sharp corners. In the two upper rows of Figure 10 we show the segmentation results for different morphological GAC regularizations. In the morphological approaches we can increase the strength of the regularization by repeating the $IS \circ SI$ operator. In the top row we show the results involving one regularization step per algorithm iteration ($\mu = 1$). In the middle row we show the evolution results when applying two regularization steps per iteration ($\mu = 2$). We can observe that the small background blobs, as well as the villi along the limbs, disappear in the strongly regularized solution. Results in Figure 10 prove that the Morphological GAC performs better with sharp and elongated structures. In terms of efficiency, the morphological approach is again one order of magnitude faster than the numerical solution.

In the following experiment we compare the evolution

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

of the energies in the morphological and numerical GAC. Since the parameters in the starfish example are different for the Morphological GAC and the standard GAC (note that $\alpha = 10^3$ in the first case and $\alpha = 10^6$ in the second one) their energies are not comparable. In order to perform a comparison between the evolution of the energies for both methods, we ran again the GAC algorithm with $\alpha = 10^3$. Figure 11(a) shows the evolution of the energy of the Morphological GAC (i.e., the evolution seen in Figure 10) and the energy of the GAC with the new parameter set. The energy for the morphological version of the algorithm is computed with the parameter $\mu = 0.1$ given in Table 1 for their continuous counterparts, since the parameter $\mu = 2$ of the morphological methods is meaningless for energy computation. Note that in both cases the global trend in the evolution tends to decrease the energy. However, in both approaches, it increases in some intervals along the evolution. This is due to both the balloon force term included in the PDE, that is not part of the minimized energy, and the fact that functional gradient descent optimization does not guarantees that the energy always decreases. Note also that the evolution of the energy for the GAC continues to fall below the minimum energy reached by its morphological counterpart. This is because the borders of the starfish limbs are almost perpendicular to the front of evolution and they are not capable of stopping the shrinkage of the curve. Hence, it continues to evolve until it becomes a point and therefore its energy is zero. It is a well-known problem of the GAC model that the global minimum of the energy is reached when the curve shrinks to a point. Therefore, the image segmentation with GAC heavily relies on the existence of local minima.

In Figures 12 and 13 we compare the performance of the morphological and numerical ACWE. In the first Figure we segment a group of lakes in a satellite image. Here both methods work in uninformative (zero-gradient) areas without the balloon force. Also, the initial curve u_0 does not require to surround the objects. We also consider the detection of point clouds in images, where the ACWE model is known to work much better than the GAC [10]. Figure 13 depicts the evolution of a curve guided by an image of Europe night lights. Again, morphological and numerical results are very similar, with the morphological algorithm outperforming the numerical one in terms of processing time (see Table 2).

The evolution of the energy for the ACWE-based methods in the lakes experiment is plotted in Figure 11(b). As above, the parameter μ for computing the energy of the morphological ACWE is the continuous one $\mu = 0.2$. The morphological parameter $\mu = 3$ is meaningless for energy computation. We observe that in the case of ACWE the energies of both methods have no oscillations and steadily decrease in each iteration. This nice behavior is due to the fact that the ACWE energy is expected to be smoother than the GAC one because of the area global terms and the lack of the balloon force

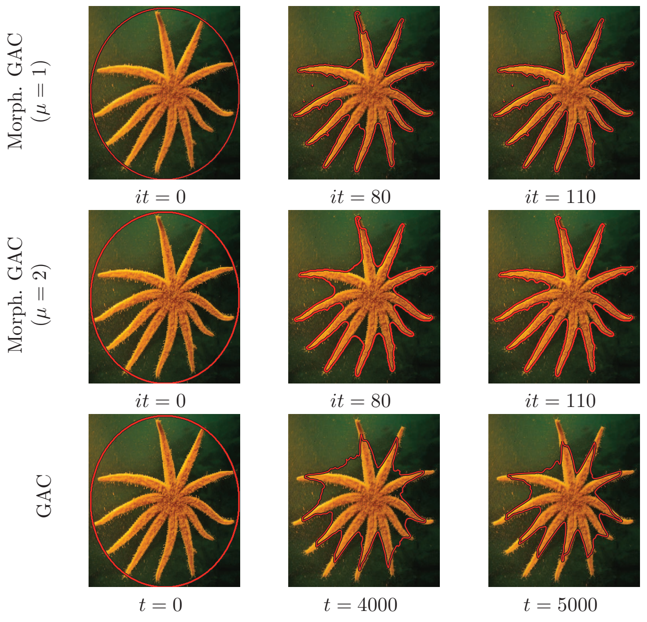

Fig. 10. Segmentation of elongated and narrow structures. GAC initialization $u_0 = 135 - \sqrt{(x-138)^2 + \frac{3}{4}(y-163)^2}$. Morphological GAC initialization $u_0 = T\left[135 - \sqrt{(x-138)^2 + \frac{3}{4}(y-163)^2} > 0\right]$.

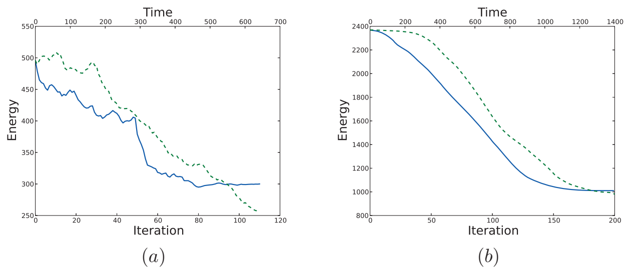

Fig. 11. Evolution of the energy of (a) the Morphological GAC and the continuous GAC in the starfish experiment and (b) the Morphological ACWE and the continuous ACWE in the lakes experiment. The blue continuous line plots the energy versus the number of iterations for the morphological methods. The green dashed line plots the energy versus the time for the continuous methods.

present in the GAC model. Moreover, they both converge to nearly the same energy value.

We also validate our model with a 3D image stack from the area of neuroscience. In this experiment we can evaluate the morphological evolution of a surface in 3D space. We use a section of cortex tissue captured with a confocal microscopy corresponding to the 3D image of a dendrite. A small 2-sphere is manually placed in the center of the image. Then, the 3D Morphological ACWE evolves the surface according to the contents of the image. Some steps of the evolution are depicted in Figure 14(a). The final results are shown in Figure 14(b).

Finally, we also make a quantitative comparison of the evolution achieved by the algorithms that we are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

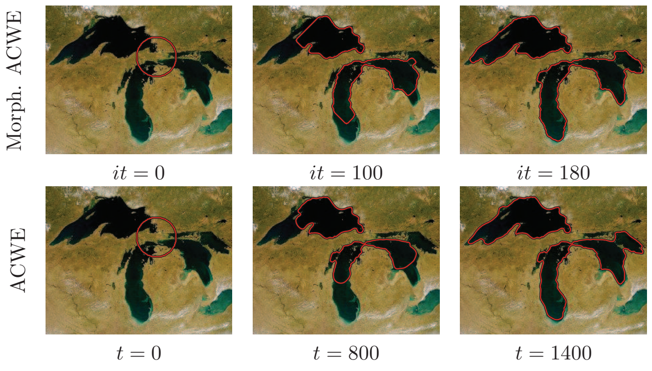

Fig. 12. Lakes segmentation. ACWE initialization $u_0 = 20 - \sqrt{(x-170)^2 + (y-80)^2}$. Morphological ACWE initialization $u_0 = T\left[20 - \sqrt{(x-170)^2 + (y-80)^2} > 0\right]$.

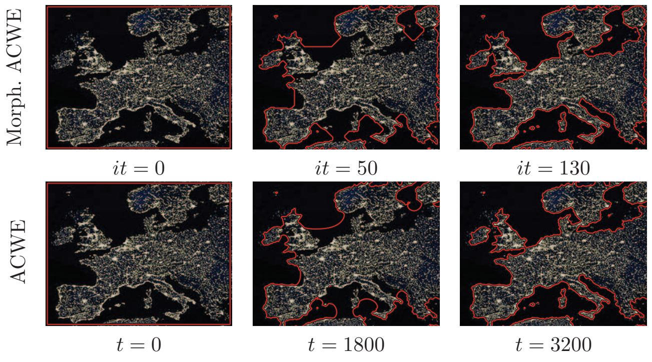

Fig. 13. Europe night-lights. ACWE initialization $u_0 = 115 - \max\{|x-120|, 1.3 \cdot |y-92|\}$. Morphological ACWE $u_0 = T[115 - \max\{|x-120|, 1.3 \cdot |y-92|\} > 0]$.

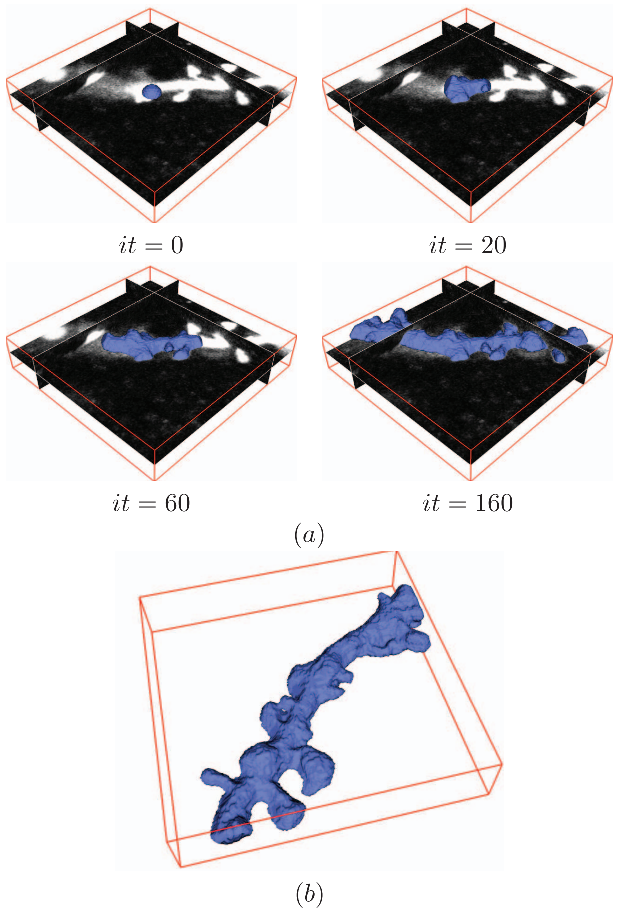

Fig. 14. Dendrite image stack. (a) Evolution of 3D Morphological ACWE. The initialization is $u_0 = T\left[10 - \sqrt{(x-20)^2 + (y-100)^2 + (z-100)^2} > 0\right]$. (c) Final result.

evaluating. We use the Jaccard similarity coefficient to quantify the comparison. Let u and v be two different contours, and χ_u and χ_v be the set of cells inside the contours, i.e., the set of cells such that $u(\mathbf{x}) > 0$ and $v(\mathbf{x}) > 0$. The Jaccard similarity between u and v is

$$J(u, v) = \frac{|\chi_u \cap \chi_v|}{|\chi_u \cup \chi_v|}. \tag{36}$$

It has value 1 when u and v are equal and 0 when they do not share any cell. Table 2 shows the similarities for the pairs of contours obtained in each experiment. They are high (around 0.9) in most cases. In the *starfish* experiment, however, the similarity is lower (0.64). An inspection of Figure 10 reveals that the morphological GAC fits more accurately the image while the numerical approach loses details at the ends of the arms and in the junctions.

6.3 Morphological turbopixels

Many computer vision algorithms work with perceptually meaningful entities of the image, obtained from a low-level grouping of pixels, called superpixels [19], [46]. Here we introduce the *morphological turbopixels* algorithm, as an example of an intrinsically local contour evolution application that provides a set of compact and regular superpixels.

Turbopixels [19] compute a dense oversegmentation of an image by growing curves from many seeds distributed throughout the image domain via a geometric flow similar to the Geodesic Active Contours. Curves emanating from different seeds are not allowed to merge into a single curve. To meet this constraint, a homotopy preserving thinning is carried out, which gives the skeleton of the area outside the curves, and partition the image domain into one cell for each curve. The curves are then confined to grow within their cell.

We have adapted the turbopixels to work with our morphological GAC, leading to the *morphological turbopixels* (see Algorithm 1). The homotopic thinning approach that we use in the steps 5 and 8 of Algorithm 1 is a type of skeletonization that preserves the topology. To this end, we use a variation of the flux-ordered thinning algorithm [47] that ignores the condition of being an endpoint. The homotopic thinning computes a mask which confines each curve in its own cell, as shown in Figure 15. After several iterations, these cells will eventually be the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

	Morphological methods		Continuous methods			
Experiment	Processing time	Iterations	Processing time	Iterations	Speedup	J
Breast nodule	$0.79s$	45	$6.85s$	268	8.67	0.87
Starfish ($\mu = 1$)	$2.46s$	110	$28.17s$	864	11.45	0.64
Lakes	$5.61s$	180	$21.90s$	867	3.9	0.95
Europe	$1.18s$	130	$17.10s$	989	14.5	0.91

TABLE 2
Running times and number of iterations of the morphological methods vs. the continuous PDEs. Last columns show the speedup gained by the morphological methods and the Jaccard similarity of the results between both methods.

superpixels. The morphological GAC with a mask behaves essentially like the standard morphological GAC, but it does not allow that the evolving curves go into the masked areas. Thus we avoid that curves growing from different seeds merge. The extension from *morphological turbopixels* to *morphological turbovoxels*, i.e., the three-dimensional counterpart, is trivial.

Algorithm 1 Morphological turbopixels

```
Input: The separation among consecutive seeds $s$, number of iterations $n$ and image $I$
1: Place seeds on a rectangular grid with step $s$
2: Perturb the seeds away from $I$ high gradient regions
3: $u \leftarrow$ binary level set with $1$ in seeds locations and $0$ everywhere else
4: for $i \in \{0, \dots, n-1\}$ do
5:     $B \leftarrow$ homotopic_thinning($u$)
6:     $u \leftarrow$ One iteration of morphological GAC of $u$ with mask $B$ and stopping criterion $g(I)$
7: end for
8: $B \leftarrow$ homotopic_thinning($u$)
Output: The boundaries of the superpixels $B$
```

Figure 16 shows some results obtained with the morphological turbopixels in two images. The top image belongs to the Berkeley database and allows qualitative comparison with the standard numerical implementation in [19]. The bottom image diplays the results for an Electron Microscopy (EM) image of a pice of brain tissue.

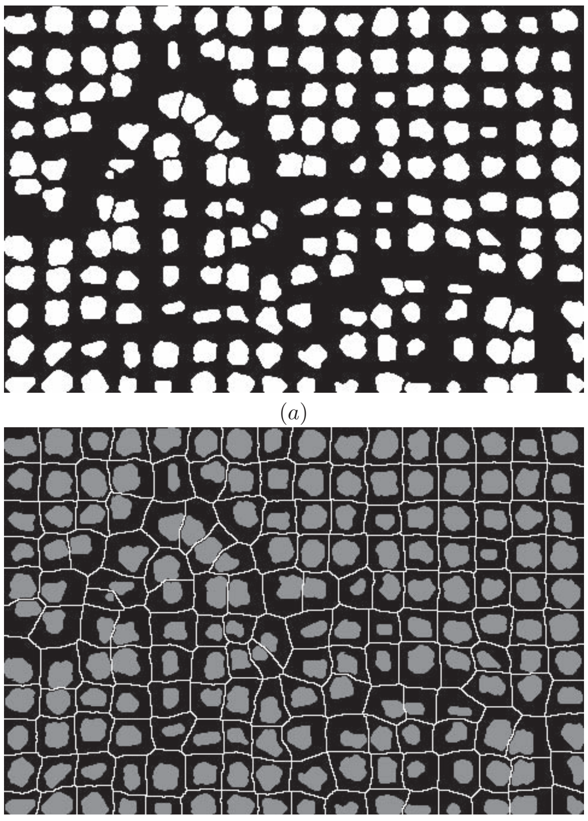

Fig. 15. (a) The level set function u describes multiple evolving curves. (b) Homotopic thinning gives a mask B that confines each curve within its own cell. The morphological GAC with mask B will evolve the curves avoiding that each curve passes through the walls of B.

7 Conclusions

This paper introduces new results relating Mathematical Morphology and PDE approaches for image analysis. We have introduced a new curvature morphological operator valid for surfaces of any dimension. On the basis of this new operator we approximate the numerical solution of surface evolution PDEs by the successive composition of morphological operators whose infinitesimal behavior is equivalent to the terms in the PDE. We have used this approach to provide morphological implementations of the *Geodesic Active Contour*, the *Active Contours Without Borders* and the *Turbopixels* algorithms. The morphological approach has several advantages over the numerical solution of the PDEs. The implementation is simpler and has fewer parameters. There are no numerical instability issues and, since the level set function is binary and does not represent a distance, it requires no re-initialization.

The experiments conducted confirm that the solutions obtained with the morphological methods are comparable to those obtained with the numerical ones, with the exception of narrow and elongated structures in which the morphological approach better fits the image. Morphological methods outperform their traditional functional gradient descent numerical counterparts in terms of stability and speed. In general, morphological

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

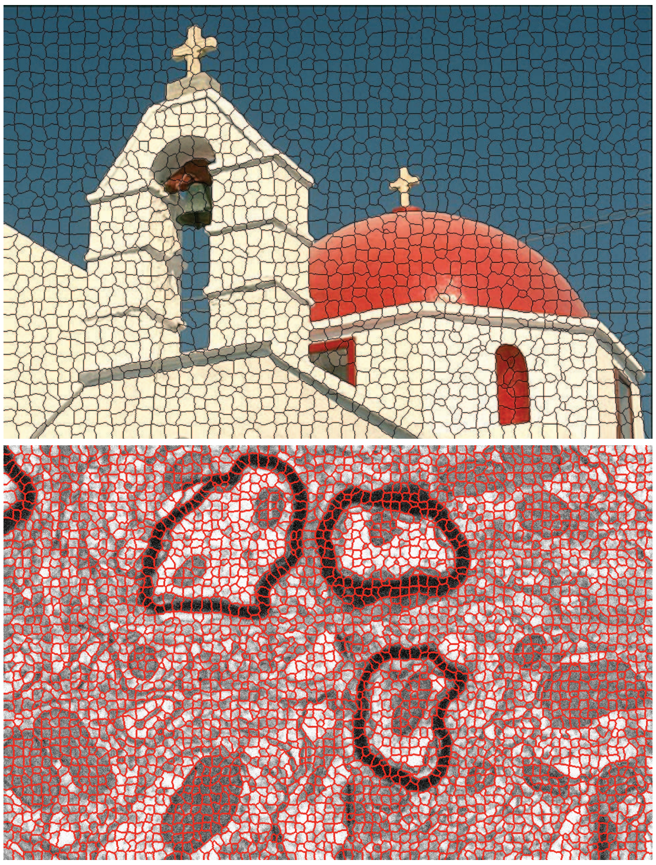
Fig. 16. Morphological turbopixels. (top) Image from the Berkeley Database; (bottom) EM image of brain tissue.

algorithms are about one order of magnitude faster, which makes them suitable for real-time applications in resource-limited hardware such as tracking in mobile devices, or for processing large images such as those from EM imaging in neuroscience applications. Moreover, the morphological approach could also benefit from improvements devised to speed-up numerical algorithms, such as the narrow band and multi-scale implementations.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and constructive suggestions. This research was funded by the Cajal Blue Brain Project.

References

[1] A. Blake and M. Isard, *Active Contours*. Springer, 1998. 1

[2] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988. 1

[3] O. Faugeras and R. Keriven, "Variational principles, surface evolution, pdes, level set methods, and the stereo problem," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 336–344, mar 1998. 1

[4] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 266–280, 2000. 1

[5] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1470–1475, August 2007. 1

[6] A. Mishra, P. Fieguth, and D. Clausi, "Decoupled active contour (dac) for boundary detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 310–324, February 2011. 1

[7] C. Zimmer and J. C. Olivo-Marin, "Coupled parametric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1838–1842, 2005. 1

[8] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for motion estimation and tracking," *Computer Vision and Image Understanding*, vol. 97, no. 3, pp. 259 – 282, 2005. 1

[9] Y. Shi and W. C. Karl, "A real-time algorithm for the approximation of level-set-based curve evolution," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 645 –656, May 2008. 1, 2

[10] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001. 1, 2, 7, 9, 13

[11] L. Vese and T. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, December 2002. 1

[12] B. Nilsson and A. Heyden, "A fast algorithm for level set-like active contours," *Pattern Recognition Letters*, vol. 24, June 2003. 1, 2

[13] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 195–215, April 2007. 1

[14] X.-F. Wang, D.-S. Huang, and H. Xu, "An efficient local chan-vese model for image segmentation," *Pattern Recognition*, vol. 43, March 2010. 1

[15] C. Lenglet, J. Campbell, M. Descoteaux, G. Haro, P. Savadjiev, D. Wassermann, A. Anwander, R. Deriche, G. Pike, G. Sapiro, K. Siddiqi, and P. Thompson, "Mathematical methods for diffusion MRI processing," *Neuroimage*, vol. 45, no. 1, pp. S111–S122, 2009. 1

[16] C. Bibby and I. D. Reid, "Robust real-time visual tracking using pixel-wise posteriors," in *Proc. European Conference on Computer Vision*, 2008, pp. II: 831–844. 1, 2

[17] P. Chockalingam, N. Pradeep, and S. Birchfield, "Adaptive fragments-based tracking of non-rigid objects using level sets," in *Proc. International Conference on Computer Vision*, 2009, pp. 1530–1537. 1, 2

[18] D. Mitzel, E. Horbert, A. Ess, and B. Leibe, "Multi-person tracking with sparse detection and continuous segmentation," in *Proc. European Conference on Computer Vision*, 2010, pp. 397–410. 1

[19] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, December 2009. 1, 2, 14, 15

[20] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997. 1, 2, 7

[21] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Gradient flows and geometric active contour models," in *Proc. International Conference on Computer Vision*, June 1995, pp. 810 –815. 1

[22] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988. 1, 3

[23] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003. 1

[24] J. Weickert, B. M. T. H. Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *IEEE Transactions on Image Processing*, vol. 7, pp. 398–410, 1998. 1

[25] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, "Fast geodesic active contours," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1467–1475, 2001. 1, 2

[26] R. Tsai and S. Osher, "Level set methods and their applications in image science," *Comm. Math Sci*, vol. 1, no. 4, pp. 1–20, 2003. 1

[27] G. Barles and P. E. Souganidis, "A New Approach to Front Propagation Problems: Theory and Applications," *Archive for Rational*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Mechanics and Analysis, vol. 141, no. 3, pp. 237–296, March 1998. 1

[28] Y. Shi and W. C. Karl, "Real-time tracking using level sets," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 34–41. 2

[29] Y. Pan, B. JD, and S. Djouadi, "Efficient implementation of the Chan-Vese models without solving PDEs," in *2006 IEEE 8th Workshop on Multimedia Signal Processing*, October 2006, pp. 350 –354. 2

[30] L. D. Cohen and R. Kimmel, "Global minimum for active contour models: A minimal path approach," *International Journal of Computer Vision*, vol. 24, no. 1, pp. 57–78, August 1997. 2

[31] B. Appleton and H. Talbot, "Globally optimal geodesic active contours," *Journal of Mathematical Imaging and Vision*, vol. 23, no. 1, pp. 67–86, July 2005. 2

[32] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, "Fast global minimization of the active contour/snake model," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 2, pp. 151–167, June 2007. 2

[33] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers, "A convex formulation of continuous multi-label problems," in *Proc. European Conference on Computer Vision*, 2008, pp. 792–805. 2

[34] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, "An algorithm for minimizing the mumford-shah functional." in *Proc. International Conference on Computer Vision*, 2009, pp. 1133–1140. 2

[35] J. Morales, L. Alonso-Nanclares, J. R. Rodriguez, J. Defelipe, A. Rodriguez, and A. Merchan-Perez, "Espina: a tool for the automated segmentation and counting of synapses in large stacks of electron microscopy images," *Frontiers in Neuroanatomy*, vol. 5, no. 18, 2011. 2

[36] P. Lax, "Numerical solution of partial differential equations," *Math. Monthly*, vol. 72, pp. 74–85, 1965. 2

[37] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel, "Axioms and fundamental equations of image processing," *Arch. Rational Mech. Anal.*, vol. 16, pp. 200–257, 1993. 2, 4

[38] R. van den Boomgaard and A. Smeulders, "The morphological structure of images: The differential equations of morphological scale-space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 11, pp. 1101–1113, November 1994. 2

[39] R. Kimmel, *Numerical Geometry of Images: Theory, Algorithms, and Applications*. Springer Verlag, 2003. 2, 3

[40] F. Guichard, J. Morel, and R. Ryan, *Contrast invariant image analysis and PDE's*, 2004. [Online]. Available: http://mw.cmla.ens-cachan.fr/~morel/JMMBookOct04.pdf 2, 3, 5

[41] F. Catté, F. Dibos, and G. Koepfler, "A morphological scheme for mean curvature motion and applications to anisotropic diffusion and motion of level sets," *SIAM Journal on Numerical Analysis*, vol. 32, no. 6, pp. 1895–1909, 1995. 2, 3, 4, 5

[42] L. Alvarez, L. Baumela, P. Henríquez, and P. Márquez-Neila, "Morphological snakes," in *Proc. International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2197 – 2202. 2

[43] L. D. Cohen, "On active contour models and balloons," *CVGIP: Image Understanding*, vol. 53, no. 2, pp. 211–218, 1991. 8

[44] M. Sussman, P. Smereka, and S. Osher, "A level set approach for computing solutions to incompressible two-phase flow," *J. Comput. Phys.*, vol. 114, pp. 146–159, September 1994. 10

[45] S. Chen, B. Merriman, S. Osher, and P. Smereka, "A simple level set method for solving Stefan problems," *J. Comput. Phys.*, vol. 135, pp. 8–29, July 1997. 10

[46] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. International Conference on Computer Vision*, vol. 1, 2003, pp. 10–17. 14

[47] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker, "Hamilton-jacobi skeletons," *International Journal of Computer Vision*, vol. 48, no. 3, pp. 215–231, July 2002. 14

Pablo Márquez-Neila received a BS in Computer Science from the Universidad de Extremadura in 2006 and a MS in Artificial Intelligence from the Universidad Politécnica de Madrid (UPM) in 2008, where he is currently completing his PhD. He has been member of the Computer Perception Group of the UPM since 2008. His research interests lie in theoretical and applied computer vision and related fields in computer graphics. He has worked in medical imaging analysis and visualization, graphical models for image processing and segmentation, and image registration.

Luis Baumela BS and MS, 1989, PhD, 1995, all in Computer Science from the Universidad Politécnica de Madrid. From 1989 to 1992 he was an engineer at Telefónica's R&D labs. Since 1997 he is Associate Professor of Computer Science at the Facultad de Informática of the Universidad Politécnica de Madrid where he leads the Computer Perception Group. His research interests include image alignment, face image analysis and medical imaging.

Luis Alvarez has received a M.Sc.in applied mathematics in 1985 and a Ph.D. in mathematics in 1988, both from Complutense University (Madrid, Spain). Between 1991 and 1992 he worked as post-doctoral researcher at CEREMADE laboratory in the computer vision research group directed by Prof. Jean-Michel Morel. Since 2000 he is full professor at the University of Las Palmas de Gran Canaria (ULPGC). He has created the research group Análisis Matemático de Imágenes (AMI) at the ULPGC. He is an expert in computer vision and applied mathematics. His main research interest areas are the applications of mathematical analysis to computer vision including problems like multiscale analysis, mathematical morphology, optic flow estimation, stereo vision, shape representation, medical imaging, synthetic image generation, camera calibration, etc.