

Active Contours on Graphs: Multiscale Morphology and Graphcuts

Kimon Drakopoulos, *Member, IEEE*, and Petros Maragos, *Fellow, IEEE*

Abstract—In this paper we propose two novel methods for formulating and implementing the methodology of geodesic active contours on arbitrary graphs, as applied to multiscale morphology and segmentation. Firstly, we propose approximations to the calculation of the gradient and the divergence of vector functions defined on graphs and use these approximations to apply the technique of geodesic active contours for object detection on graphs. To this end, we extend existing work on graph morphology to multiscale dilation and erosion and implement them recursively using level sets of functions defined on the graph. Second, we propose a graphcut based solution to the geodesic active contour problem on graphs. Appropriate weights are calculated for each edge for which the Riemannian length of a contour can be approximated by the weighted sum of intersections of the contour with the edges of the graph. Finding the minimum Riemannian length contour then becomes equivalent to solving a max flow problem for which efficient solutions have been proposed in the literature.

Index Terms—Geodesic active contours, graphcuts, image analysis, image edge detection, image segmentation, morphological operations, object detection.

I. INTRODUCTION

ACTIVE contours, both as a methodology based on variational calculus and partial differential equations (PDEs) as well as a collection of numerical geometry algorithms, have become useful for a variety of problems in image processing and computer vision, [24]. An optimized version of them, i.e., the geodesic active contours (GACs) [10] contains as a special case the PDE that can generate multiscale isotropic flat dilations and erosions [1], [8]. Further, by a proper selection of their edge-stopping function, their segmentation action becomes very close to the PDE-based active contour implementation of watershed segmentation [26]. In this paper we focus on a novel extension of GACs to morphological scale-space filtering and segmentation on *graphs*. Graph-theoretic approaches have become commonplace in image analysis and computer vision. Examples include the statistical modeling of digital images with MRFs and the graphcut approaches to segmentation [6], [23].

Manuscript received February 07, 2012; revised July 02, 2012; accepted August 08, 2012. Date of publication August 17, 2012; date of current version October 12, 2012. The main part of this work was performed while K. Drakopoulos was at the National Technical University of Athens. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Junior Barrera.

K. Drakopoulos is with the Department of Electrical and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: kimondr@mit.edu).

P. Maragos is with the School of Electrical and Computer Engineering, National Technical University of Athens, Athens 15773, Greece (e-mail: maragos@cs.ntua.gr).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTSP.2012.2213675

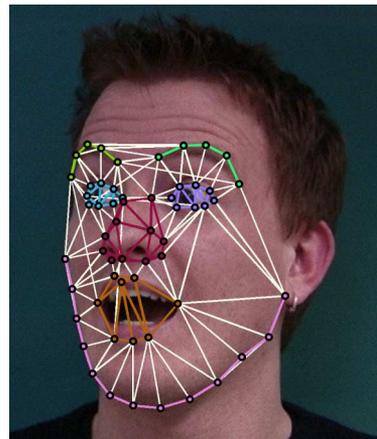


Fig. 1. Representation and segmentation of visual face information on graphs.

In most of these cases, the image graphs are regular grids that result from uniform sampling of continuous space. However, there also exist many types of visual data defined on *arbitrary graphs* with irregular spacing among their vertices. Examples from vision include the graph representing an active appearance model of a face in Fig. 1 and its subgraphs representing local models of eyes, mouth and nose. Non-visual examples include network problems modeled with graphs, such as social nets and geographical information systems.

The major contribution of this paper is to explore theoretically and algorithmically GACs on *arbitrary graphs* and their application to object segmentation. We use two different techniques to implement these: one through approximations to the calculation of vector calculus operators on graphs and one involving an explicit combinatorial optimization problem formulated using the notions of graphcuts and geocuts from [6]. A second related contribution is the development of morphological scale-space filtering on graphs, theoretically by using the lattice algebra of morphological operators on graphs and algorithmically as a special case of the GACs approximated by difference equations.

The rest of the paper is organized as follows. In Section II we begin with some definitions from lattice-based morphology. Then, we review 1) multiscale morphological image operators on a Euclidean domain, either defined algebraically or generated by nonlinear PDEs, and 2) lattice-theoretic morphological operators on graphs which have been introduced in [19], [33] with some recent work in [16]. Finally, we connect these two areas and define multiscale morphological operators on graphs.

In Section III we focus on our major approach which is based on discretizing the PDEs moving geodesic active contours [10] on arbitrary graphs and, as a special case, the PDEs generating continuous-scale morphological operators [1], [8]. In the latter

direction, a first approach to approximate morphological operators on graphs through mimicking the corresponding PDEs has been studied in Ta *et al.* [32]. Our approach is slightly different in our translation of the continuous gradient operator on arbitrary graph structures and in our usage of multiscale neighborhoods. In the general field of approximating PDE-type problems on weighted graphs, a systematic analysis has been performed in [3], [12] by introducing discrete gradients and Laplacian and by studying Dirichlet and Neumann boundary value problems on graphs. Moreover, extensive work has been done on discrete calculus ([17], [20]), where instead of discretizing the conventional calculus, an independent theory is developed that operates purely in the discrete space.

Finally, in Section IV we formulate an explicit combinatorial minimization problem corresponding to the minimization of the geodesic length of a contour on a graph. The latter can be written in terms of finding the minimum cut on a graph with modified edge weights. Our contribution lies in the calculation of the appropriate edge weights in such a way that makes the minimum cut problem and the minimum length contour problem approximately equivalent through Cauchy-Crofton type equations. This idea has first been analyzed for regular grids in [4], [6] and efficient methods for solving the associated combinatorial optimization problem have been studied in [5], [23] and [11] for a similar application. We extend these ideas to arbitrary graphs which involves a delicate treatment of the Cauchy-Crofton formulas and use gradient-descent type methods to solve the optimization problem which allows us to observe the actual evolution of the curves.

II. MULTISCALE AND GRAPH MORPHOLOGY

A. Background on Lattice Morphology

A *complete lattice* is a partially ordered set (\mathcal{L}, \leq) where, for any subset \mathcal{K} of the lattice \mathcal{L} , its supremum $(\bigvee \mathcal{K})$ and infimum $(\bigwedge \mathcal{K})$ exist in \mathcal{L} . The algebraic formalization [18], [31] of morphological operators views them as operators on complete lattices and unifies their usage for both image and shape analysis. Two main examples of complete lattices used in morphological analysis are: (i) the power set $\mathcal{P}(E) = \{X : X \subseteq E\}$ of all binary images or shapes represented by subsets X of some domain E where the \bigvee / \bigwedge lattice operations are the set union/intersection, and (ii) the space of all graylevel image signals $f : E \rightarrow \mathcal{T}$ with values in a sublattice \mathcal{T} of $\mathbb{R} = \mathbb{R} \cup \{-\infty, \infty\}$ and the \bigvee / \bigwedge lattice operations are the supremum/infimum of sets of real numbers. An operator ψ on \mathcal{L} is called *increasing* if it preserves the partial ordering \leq . Four important types of increasing operators are the dilations (δ), erosions (ε), openings (α) and closings (β). Dilations and erosions distribute respectively over the supremum and infimum of any collection of input signals. They are the fundamental blocks for building composite morphological systems. Openings (resp. closings) are increasing, anti-extensive (resp. extensive) and idempotent; they play the role of nonlinear smoothing filters. Dilations and erosions come in pairs (δ, ε) called *adjunctions* if

$$\delta(f) \leq g \iff f \leq \varepsilon(g). \quad (1)$$

Such pairs are useful for constructing openings $\alpha = \delta\varepsilon$ and closings $\beta = \varepsilon\delta$. Operator products mean composition: $\phi\psi(f) = \phi(\psi(f))$. The notation ψ^r means r -fold composition.

B. Euclidean Multiscale Morphology

In Euclidean morphology, the domain E becomes the d -dimensional space \mathbb{E}^d where $\mathbb{E} = \mathbb{R}$ or $\mathbb{E} = \mathbb{Z}$. The most well-known morphological operators, which are simple special cases of their lattice counterparts, are the translation-invariant (TI) Minkowski dilation $X \oplus B = \bigcup_{\mathbf{b} \in B} X_{+\mathbf{b}}$ and Minkowski erosion $X \ominus B = \bigcap_{\mathbf{b} \in B} X_{-\mathbf{b}}$ of two subsets X and B of \mathbb{E}^d , where $X_{+\mathbf{b}} = \{\mathbf{x} + \mathbf{b} : \mathbf{x} \in X\}$ denotes the translation of $X \subseteq \mathbb{E}^d$ by $\mathbf{b} \in \mathbb{E}^d$. In the continuous case, by scaling the structuring set B via a homothety $rB = \{r\mathbf{b} : \mathbf{b} \in B\}$ where $r \geq 0$, we can define *multiscale* TI morphological set operators on \mathbb{R}^d :

$$\delta_{rB}(X) \triangleq X \oplus rB, \quad \varepsilon_{rB}(X) \triangleq X \ominus rB. \quad (2)$$

Similarly, if $(f \oplus B)(\mathbf{x}) = \bigvee_{\mathbf{b} \in B} f(\mathbf{x} - \mathbf{b})$ and $(f \ominus B)(\mathbf{x}) = \bigwedge_{\mathbf{b} \in B} f(\mathbf{x} + \mathbf{b})$ are the unit-scale Minkowski *flat* function operators, their multiscale counterparts are

$$\delta_{rB}(f) \triangleq f \oplus rB, \quad \varepsilon_{rB}(f) \triangleq f \ominus rB. \quad (3)$$

If B is *convex*, the above multiscale set and function operators obey a *semigroup* property

$$\delta_{(r+1)B} = \delta_B \delta_B^r, \quad \varepsilon_{(r+1)B} = \varepsilon_B \varepsilon_B^r, \quad r = 0, 1, 2, \dots, \quad (4)$$

because rB equals the r -fold dilation of B with itself [27]

$$rB = \underbrace{B \oplus B \oplus \dots \oplus B}_{r \text{ times}}, \quad r = 0, 1, 2, \dots \quad (5)$$

The above ideas create the simplest case of a *morphological scale-space* [26].

The above TI multiscale morphological operators can be extended to *digital shapes and images* by using two alternative approaches. The first is an algebraic approach where if $B \subseteq \mathbb{Z}^d$ a unit-scale discrete structuring set, we define its scaled version rB for integer scales r as its r -fold self-dilation in (5) and use (4) for producing multiscale morphological operators that agree with their sampled continuous versions if B is convex. The second approach [1], [8] models the dilation and erosion scale-space functions $u(\mathbf{x}, t) = f(\mathbf{x}) \oplus tB$ and $v(\mathbf{x}, t) = f(\mathbf{x}) \ominus tB$ as generated by the nonlinear PDEs

$$\partial_t u = \|\nabla u\|_B, \quad \partial_t v = -\|\nabla v\|_B \quad (6)$$

where for a convex $B \subseteq \mathbb{R}^2$, $\|(x_1, x_2)\|_B = \sup_{(a_1, a_2) \in B} a_1 x_1 + a_2 x_2$. These PDEs can be implemented using the numerical algorithm of [29], as explored in [30]. In case of a shape X , the above PDEs can still be used to generate its multiscale morphological evolutions by treating u as the level function whose zero level set contains the evolving shape. Such PDE-based shape evolutions have been studied in [22]. Modern numerical algorithms for morphological PDEs can be found in [7].

C. Background on Graph Morphology

We consider an undirected **graph** $G = (V, E)$ without self-referring loops and multiple edges, where $V = V(G)$ and $E =$

$E(G)$ are the sets of its vertices (also called nodes) and edges, respectively. We denote edges by pairs (v, w) of vertices; these are symmetric, i.e., $(v, w) = (w, v)$, since the graph is undirected. If $V' \subseteq V$ and $E' \subseteq E$ with $E' \subseteq V' \times V'$, the pair $G' = (V', E')$ is called a *subgraph* of G . A graph vertex mapping $\theta : V \rightarrow V'$ is called a *graph homomorphism* from G to G' if θ is one-to-one and preserves edges, i.e., $(v, w) \in E$ implies $(\theta(v), \theta(w)) \in E'$. If θ is a bijection, then it is called a *graph isomorphism*; if in addition $G' = G$, then it is called a *graph automorphism* or *symmetry* of G . The set of all such symmetries forms under composition the *symmetry group* $\text{Sym}(G)$ of a graph. Symmetries play the role of ‘generalized translations’ on a graph.

Shapes $X \subseteq V$ and image functions $f : V \rightarrow \mathcal{T}$ defined on a graph G with values in a complete lattice \mathcal{T} will be denoted by $(X|G)$ and $(f|G)$, respectively, and may be referred to as *binary graphs* and *multilevel graphs*. In case of multilevel graphs, the values of the functions $(f|G)$ may be discrete, e.g., $\mathcal{T} = \{0, 1, \dots, m-1\}$, or continuous, e.g., $\mathcal{T} = \overline{\mathbb{R}}$. Similarly a graph operator for shapes or functions will be denoted by $\psi(\cdot|G)$. The argument G will be omitted if there is no risk of confusion. A graph operator ψ is called *increasing* if it is increasing in its first argument, i.e., $X \subseteq Y$ implies $\psi(X|G) \subseteq \psi(Y|G)$. A graph operator ψ is called *invariant* under graph symmetries $\tau \in \text{Sym}(G)$ if $\tau\psi = \psi\tau$.

Henceforth and until mentioned otherwise, we shall focus our discussion on binary graph operators. Given a graph $G = (V, E)$, the binary graph dilations and erosions on $\mathcal{P}(V)$ can be defined via a **graph neighborhood function** $N : V \rightarrow \mathcal{P}(V)$ which assigns at each vertex v a neighborhood $N(v)$. Taking the union of all such neighborhoods for the vertices of a shape $X \subseteq V$ creates a graph dilation of X ; then, by using (1) we also find its adjunct erosion:

$$\begin{aligned} \delta_N(X|G) &\triangleq \bigcup_{v \in X} N(v), \\ \varepsilon_N(X|G) &\triangleq \{v \in V : N(v) \subseteq X\}. \end{aligned} \quad (7)$$

At each vertex v , the shape of $N(v)$ may vary according to the local graph structure and this inherently makes the above morphological graph operators *adaptive*. At each v , the *reflected neighborhood* is defined by

$$\tilde{N}(v) \triangleq \{w \in V : v \in N(w)\}. \quad (8)$$

Then, the dual operations (w.r.t. complementation) of graph dilation and erosion w.r.t. a neighborhood coincide with the erosion and dilation, respectively, w.r.t. the reflected neighborhood. If $N(v) = \tilde{N}(v)$ for each v , we have a *symmetric neighborhood function*. Such an example is Vincent’s *unit-scale graph neighborhood function* [33]

$$N_1(v) \triangleq \{w \in V : (v, w) \in E\} \cup \{v\} \quad (9)$$

which, when centered at a vertex v , includes this vertex and all others that form an edge with it. If we use it in (7), this leads to the simplest unit-scale graph dilation $\delta_1(X|G)$ and erosion $\varepsilon_1(X|G)$. Since $(\delta_1, \varepsilon_1)$ is an adjunction, the composition

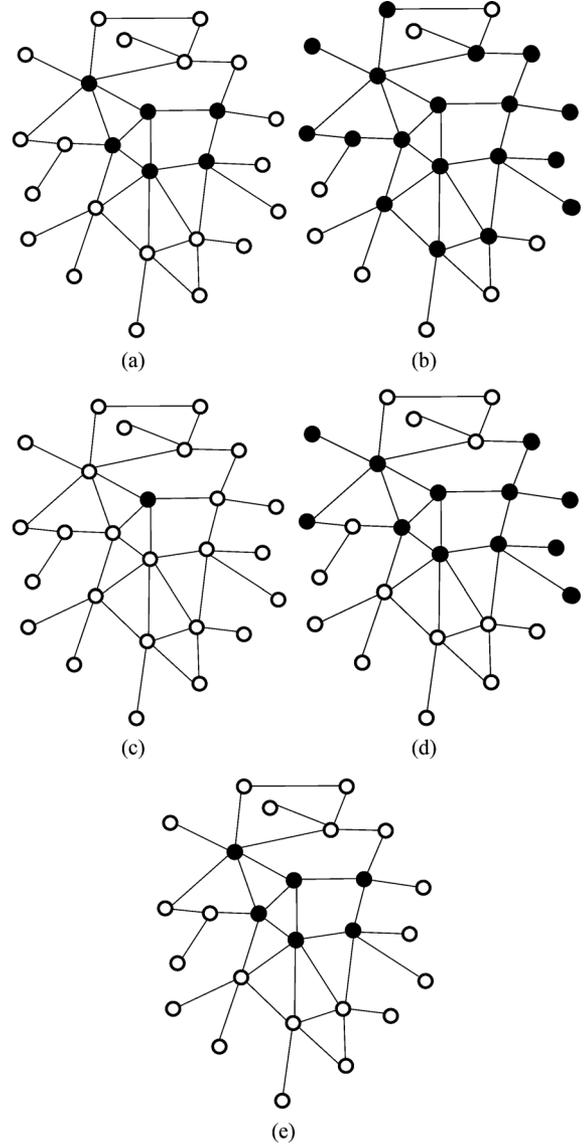


Fig. 2. Binary graph operators using a unit-scale symmetric neighborhood function (a) The vertex set on which we apply morphological operators (b) Dilation (c) Erosion (d) Closing (e) Opening.

$\alpha_1 = \delta_1 \varepsilon_1$ and $\beta_1 = \varepsilon_1 \delta_1$ is a graph opening and closing, respectively. See Fig. 2 and Fig. 5 for an example. All four of these operators inherit the standard increasing property from their lattice definition and are invariant under graph symmetries.

Heijmans *et al.* [19] have generalized the above (symmetric neighborhood N_1) approach by introducing the concept of a **structuring graph** (*s-graph*). This is a graph $A = (V_{\mathcal{A}}, E_{\mathcal{A}})$ of a relatively small size and has as additional structure two nonempty and possibly overlapping subsets: the *buds* $B_{\mathcal{A}} \subseteq V_{\mathcal{A}}$ and the *roots* $R_{\mathcal{A}} \subseteq V_{\mathcal{A}}$. It may not be connected and plays the role of a locally adaptive graph template, where (compared to Euclidean morphology) the buds correspond to the points of a structuring graph and the roots correspond to its origin. See Fig. 3 for examples. An s-graph \mathcal{A} corresponds to the following neighborhood function

$$N_{\mathcal{A}}(v|G) = \bigcup \{\theta(B_{\mathcal{A}}) : \theta \text{ embeds } \mathcal{A} \text{ into } G \text{ at } v\}. \quad (10)$$

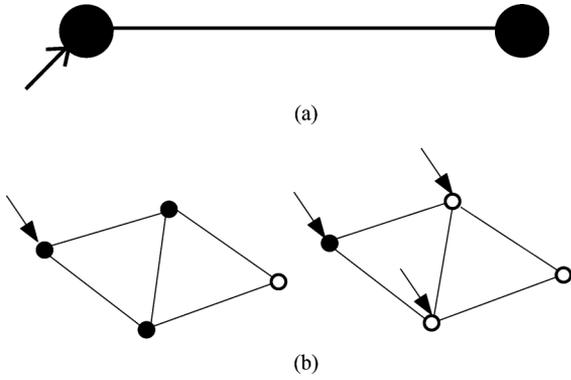


Fig. 3. Examples of structuring graphs. Arrows indicate roots, disks denote buds and circles denote nodes that do not fall in any of the above categories (a) The s-graph that corresponds to the simple neighborhood. Specifically, using this s-graph as a structuring element, the neighborhood of a node is the set of nodes that are adjacent to it (b) A structuring graph and its reflection. The reflected s-graph has the same vertices and edges as the original s-graph but their bud and root sets are interchanged.

where we say that θ embeds \mathcal{A} into G at v if θ is a graph homomorphism of A into G and $v \in \theta(R_{\mathcal{A}})$. Such an embedding matches the s-graph \mathcal{A} with the local structure of the graph G . The simple neighborhood N_1 of (9) corresponds to the s-graph of Fig. 3(a), with two vertices which both are buds and one of them is a root. Replacing (10) in (7) creates an adjunction of graph dilation and erosion $(\delta_A, \varepsilon_{\mathcal{A}})$ by structuring graphs. These are *symmetry-invariant* operators, i.e., they commute with group symmetries τ , because the neighborhood function of their s-graph is invariant under group symmetries: i.e., $N_{\mathcal{A}}(\tau(v)|G) = \tau N_{\mathcal{A}}(v|G)$, where $\tau X = \{\tau(v) : v \in X\}$.

Finally, the reflection of the neighborhood of an s-graph equals the neighborhood of another s-graph \mathcal{A} , called the *reflection* of \mathcal{A} , which has the same vertices and edges as the original s-graph \mathcal{A} but their bud and root sets are interchanged (see Fig. 3).

All the previously defined binary graph operators are increasing and can be extended to multilevel graphs. Specifically, a multilevel graph $(f|G)$ can also be represented by its *levelsets* $X_h(f|G) = \{v \in V : f(v) \geq h\}$, $h \in \mathcal{T}$:

$$(f|G)(v) = \sup\{h \in \mathcal{T} : v \in X_h(f|G)\}. \quad (11)$$

By applying an increasing binary graph operator ψ to all level sets and using threshold superposition, we can extend ψ to a *flat operator* on multilevel graphs:

$$\psi(f|G)(v) = \sup\{h \in \mathcal{T} : v \in \psi(X_h(f)|G)\}. \quad (12)$$

For example, if $\psi(X|G)$ is a set dilation by the s-graph \mathcal{A} , the corresponding function operator is

$$\delta_{\mathcal{A}}(f|G)(v) = \max_{w \in N_{\mathcal{A}}(v|G)} f(w). \quad (13)$$

Two choices for the function values are either discrete with $\mathcal{T} = \{0, 1, \dots, m-1\}$, or continuous with $\mathcal{T} = \mathbb{R}$.

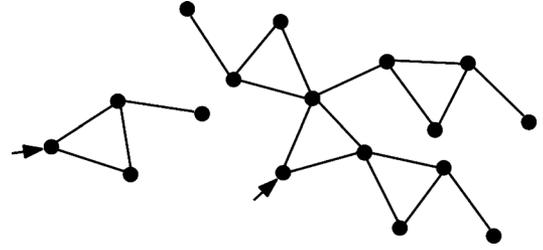


Fig. 4. Left: a structuring graph. Right: The scaled by $r = 2$ version of the s-graph. A scaling of a simple s-graph has increasingly complicated structure and therefore, for larger scales it is difficult or impossible to find an embedding to an arbitrary graph at each node. This fact necessitates an alternative definition of scale on graphs.

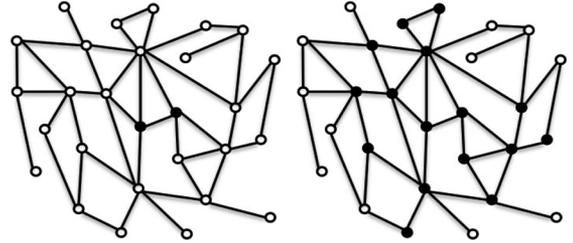


Fig. 5. Dilation of the subgraph on the left with the unit-scale structuring graph of Fig. 4.

D. Multiscale Morphology on Graphs

In order to obtain the graph counterparts of multiscale dilation and erosion, we need to define a concept of scale in a graph. Consider a graph $G = (V, E)$ and a nonempty subset $X \subseteq V$ of its vertices. Let \mathcal{A} be an s-graph. One approach could be to define the dilation at scale $r = 1, 2, \dots$ of a vertex subset X w.r.t. the s-graph \mathcal{A} by $\delta_{r, \mathcal{A}}(X|G)$ where r, \mathcal{A} denotes the r -fold graph dilation of the s-graph with itself on an underlying graph that consists of copies of A . This approach would encounter the problem presented in Fig. 4. Specifically the scaled versions of the s-graph have complicated structure and in general, it would be highly unlikely to find an appropriate embedding in the graph at every node of X to calculate the dilation of the set.

Thus, we propose the following alternative definition of multiscale dilation and erosion in order to overcome the issues mentioned. We define recursively the graph dilation of X at integer scale $r = 1, 2, \dots$ with respect to the s-graph \mathcal{A} by

$$\delta_{\mathcal{A}}^r(X|G) = \delta_A(\delta_{\mathcal{A}}^{r-1}(X|G)|G). \quad (14)$$

Thus, instead of first finding an r -scaling of the structuring graph and then performing the dilation with the original set, we recursively dilate the set X with the unit-scale structuring graph r times. Generalizing this notion of scale to multilevel dilation of a function $f : V \rightarrow \mathcal{T}$ we get the following definition. The dilation of f at integer scales r will be given at each $v \in V$ by

$$\begin{aligned} \phi_r(v) &\triangleq \delta_{\mathcal{A}}(\phi_{r-1}(f|G)|G)(v), \\ \phi_1(v) &= \delta_{\mathcal{A}}(f|G)(v). \end{aligned} \quad (15)$$

This provides a recursive computation of the scaled dilations of a function $f : V \rightarrow \mathcal{T}$ and leads us to the following Proposition which offers an alternative recursive computation that involves a simple morphological gradient on a graph.

Proposition 1: Given a graph $G = (V, E)$, the evolution of the multiscale dilation of a function $f : V \rightarrow \mathcal{T}$ by an s -graph \mathcal{A} is described by the following difference equation at each vertex $v \in V$:

$$\phi_{r+1}(v) - \phi_r(v) = \max_{w \in N_{\mathcal{A}}(v|G)} \{\phi_r(w) - \phi_r(v)\}. \quad (16)$$

Proof: By combining (15) with (13). \blacksquare

III. GEODESIC ACTIVE CONTOURS ON GRAPHS

Active contours have long been used in the computer vision community and were first introduced by Kass *et al.* in [21] in the form of ‘snakes.’ They can be described as deformable curves driven by image based forces towards features like edges and lines. Typically they are used to find in an image areas that are naturally distinguished from their background. Originally, they were implemented in a brute-force way by minimizing between successive time steps the energy functionals associated with the current state of the contour; its’ shape and distance from the salient image features.

This approach has many drawbacks: 1) It involves solving a minimization problem at each time step and 2) the curve dynamics incurred do not allow changing the topology of the original contour; for example if the original curve contains two distinct objects the initial contour will not be separated in two independent evolving contours. The latter is considered as one of the main limitations of the original approach and many heuristic solutions have been proposed [28] but a topology-free approach has been given independently by Caselles *et al.* [9] and Malladi *et al.* [25].

These solutions proposed describe the evolution of a curve that is propagating by means of a velocity that contains two terms, one related to the regularity of the curve and the other shrinks or expands towards the desired image feature. The curve dynamics can be described by a PDE for which the level set methods proposed by Osher and Sethian [29] provide an efficient implementation framework that can accommodate changes in the curve’s topology.

In particular let $\mathbf{C}(q) : [0, 1] \rightarrow \mathbb{R}^2$ be a parametrized planar curve and let $I : [0, 1]^2 \rightarrow \mathbb{R}_+$ be the image in which one needs to detect the objects’ boundaries. (We denote the curve by $\mathbf{C}(\cdot)$ when we interpret it as a vector and by C when we interpret it as a set of points in the Euclidean space.) The energy functional associated with C can be written as follows ($\alpha, \lambda \geq 0$):

$$E(C) = \alpha \int_0^1 \|\mathbf{C}'(q)\|^2 dq - \lambda \int_0^1 g(\|\nabla I(\mathbf{C}(q))\|) dq \quad (17)$$

where $g : [0, +\infty) \rightarrow \mathbb{R}_+$ is a strictly decreasing function that tends to zero as $r \rightarrow \infty$.

As shown by Caselles *et al.* in [10], finding the curve that minimizes (17) is equivalent to finding the minimum length curve

in a Riemannian space induced from image I , where length is given by

$$L_R = \int_0^1 g(\|\nabla I(\mathbf{C}(q))\|) \|\mathbf{C}'(q)\| dq \quad (18)$$

$$= \int_0^{L(C)} g(\|\nabla I(\mathbf{C}(s))\|) ds, \quad (19)$$

where $L(C)$ is the Euclidean length of the curve C and s is arclength. Furthermore, they showed that a curve that is governed from the dynamics

$$\frac{\partial \mathbf{C}(t)}{\partial t} = g(I) \kappa \mathbf{N} - (\nabla g \cdot \mathbf{N}) \mathbf{N} \quad (20)$$

where κ is the Euclidean curvature and \mathbf{N} is the unit inward normal, moves in the direction of the gradient of the length L_R . In practice, we often add a balloon force, i.e., a curve motion of the type $\partial_t \mathbf{C} = g(I) c \mathbf{N}$ where c is a constant velocity.

If we write the curve $C(t)$ as the level set of a function $u : \mathbb{R}^2 \times \mathbb{R}_+ \rightarrow \mathbb{R}$, that is write $C(t)$ as the set of points \mathbf{x} for which $u(\mathbf{x}, t)$ is equal to a constant (for example $u = 0$) then if the function $u(\mathbf{x}, t)$ satisfies

$$\frac{\partial u}{\partial t} = g(I) \|\nabla u\| (c + \kappa) + \nabla g(I) \cdot \nabla u, \quad (21)$$

the corresponding level set satisfies (20) after we add to it a constant velocity term.

The rest of this section and our main contribution in the level set implementation of active contours on graphs is to approximate all terms in the right hand side of (21) on graphs and finally construct a difference equation which would approximate the active contour dynamics for object boundary detection.

We start in the next subsection with the simplest case of curve evolution on graphs: constant velocity motion. We use useful ideas from graph morphology operators by exploiting the underlying connection between constant rate expansion and dilation of a shape. This case corresponds to approximating the first term of the right hand side of (21). Subsequently, we approximate the rest of the terms participating in the curve dynamics to end up with a geodesic active contour model on graphs.

A. Constant-Velocity Active Contours on Graphs

In this subsection we start by deriving the difference equation that describes the evolution of a contour that expands with constant velocity on a graph.

Experience from the continuous space analysis suggests that if the contour of a set expands with constant velocity then it can be described in terms of the boundary of its scaled dilation with respect to a unit ball. If the graylevel function $u : \mathbb{R}^2 \times \mathbb{R}_+ \rightarrow \mathbb{R}$ embeds the expanding contour as a level set, then the evolution of u can be described by

$$\frac{\partial u}{\partial t} = \|\nabla u\|. \quad (22)$$

We adopt the same idea on an arbitrary discrete structure, a graph $G = (V, E)$ where V denotes the set of vertices and E the set of edges. Let $X \subseteq V$ and \mathcal{A} be a structuring graph. As

for the continuous case, the constant velocity expansion of X corresponds to its r -scaled dilation, denoted by X_r .

Let $u_0 : X \rightarrow \mathbb{R}$ and $u_r : X \rightarrow \mathbb{R}$ be the graylevel functions that embed X and X_r respectively as level sets. Then the difference equation that governs u_r , by using proposition 1, is

$$u_{r+1}(v) - u_r(v) = \max_{w \in N_{sf}(v|G)} \{u_r(w) - u_r(v)\}. \quad (23)$$

The above expression is a discrete equivalent of the gradient magnitude on graphs. Similar expressions are being used in the literature, e.g., [3], [12], [32]. From a calculus perspective this expression assumes homogeneity of the underlying graph since it equals the maximum difference in values and not the maximum rate of increase as suggested by the standard gradient definition.

In order to account for topological inhomogeneities of the graph one could calculate the gradient as the maximum rate of increase and its direction as the direction of the edge along which the rate of increase is larger, implying the approximation of $\|\nabla u\|$ at node v by

$$\max_{w \in N_{sf}(v|G)} \left\{ \frac{u(w) - u(v)}{d(w, v)} \right\}, \quad (24)$$

where $d(w, v)$ is the Euclidean distance between the nodes w and v . In applications where the underlying graph is reasonably homogeneous (for example the geometric random graph where nodes are uniformly distributed around their neighbors) the two approximations presented above behave similarly. On the other hand, the latter need to be taken into consideration when proving the convergence results of Section III-F since otherwise, in the limit of large graphs the first approximation yields zero by continuity of the underlying function.

Summarizing, let X be a set of nodes whose contour expands with constant velocity parametrized by c . Then in order to implement its evolution on a graph one should proceed as follows:

- (i) Let u_0 be the signed distance function from X , defined by

$$u_0(v) = \begin{cases} \min_{w \in G \setminus X} d(w, v) & \text{if } v \in X, \\ -\min_{w \in X} d(w, v) & \text{if } v \notin X, \end{cases}$$

where X is the zero level set of u_0 .

- (ii) Evolve u_r according to (23) at scales $r = 1, 2, \dots$
- (iii) The set $X_r = \{v \in X : u_r(v) \geq 0\}$ corresponds to the r -scaled dilation of X .

Fig. 6 illustrates the results (as level sets) for the constant velocity expansion of an orthogonal contour. Throughout this paper for all our simulations we are using the simple structuring graph of Fig. 3(a) that generates the simple neighborhood. The underlying graph is a *geometric random graph* on the unit square. The geometric random graph is characterized by two parameters; the number of nodes N and a radius ρ . N nodes are being placed uniformly and at random on the unit square independently from one another. If the Euclidean distance between those two nodes is less than ρ then there is an edge between them. Typical values for N is 3000 to 10000 while ρ ranges from 0.015 to 0.04. Given the number of nodes, the parameter ρ

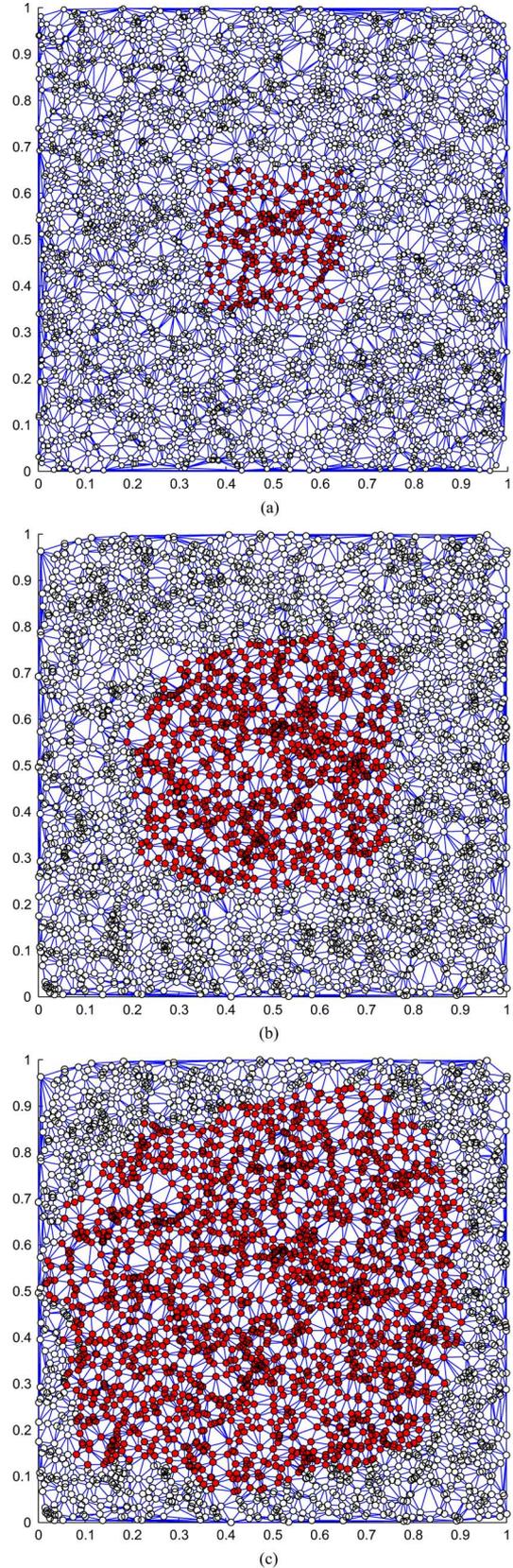


Fig. 6. Constant velocity evolution of an orthogonal contour on a geometric random graph on the unit square. The structuring graph is an edge (a) Original object (b) Dilation after 4 iterations (c) Dilation after 12 iterations.

affects the expected degree of a node and is proportional to the square root of its value.

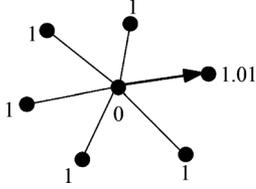


Fig. 7. Illustration of the disadvantages of choosing the maximum increase direction as the gradient direction. Observe that all directions yield approximately the same increase but only one edge is chosen. This happens exactly because of the discrete nature of the graph structure. Alternatively, we propose a weighted average of edge directions where the weights are the function differences along each edge.

B. Direction of the Gradient on Graphs

The next term from (21) that we should approximate on a graph is the direction of the gradient vector. Trusting our intuition from calculus, it would make sense to choose as the direction of the gradient on graphs the direction of the edge that corresponds to the maximum difference of the values of the function u . Namely, let a function u be defined on the set of graph nodes $v \in V$. Then

$$\frac{\nabla u}{\|\nabla u\|}(v) = \mathbf{e}_{v\hat{w}},$$

$$\hat{w} = \arg \max_{w \in N_{\text{adj}}(v|G)} \{u(w) - u(v)\}. \quad (25)$$

where \mathbf{e}_{vw} is the unit vector in the direction of the edge (v, w) .

The above approximation, assigns to the gradient the direction of the maximum change among all edges adjacent to a node. In Section III-F we prove that for the case of geometric random graphs this expression converges in probability to the true value of the gradient on each node. Note that we assume homogeneity on the graph. In any other case one can choose the direction of the maximum rate of change namely,

$$\frac{\nabla u}{\|\nabla u\|}(v) = \mathbf{e}_{v\hat{w}},$$

$$\hat{w} = \arg \max_{w \in N_{\text{adj}}(v|G)} \left\{ \frac{u(w) - u(v)}{d(w, v)} \right\}. \quad (26)$$

This approximation encounters serious drawbacks in practice which are described in Fig. 7. Such a scenario is fairly common in a graph structure due to its discrete nature. In words, choosing an edge out of finitely many can create asymmetries which vastly influence the result. Specifically, by using the edge indicated in the figure for our calculations neglects the values of the function for the rest of the nodes in one's neighborhood. Note that in the continuous case such an issue does not occur under the continuity and differentiability assumptions that are usually made.

A heuristic fix to the above issue is the approximation of the gradient direction on graphs through a weighted average of the direction of all edges in a node's neighborhood. The weights are chosen as the normalized edge differences, that is

$$\frac{\nabla u}{\|\nabla u\|}(v) \triangleq \frac{\sum_{w \in N_{\text{adj}}(v|G)} (u(w) - u(v)) \mathbf{e}_{vw}}{\left\| \sum_{w \in N_{\text{adj}}(v|G)} (u(w) - u(v)) \mathbf{e}_{vw} \right\|}. \quad (27)$$

In many applications one may need to further suppress the inherent inhomogeneity due to the sparsity of the underlying discrete structure. In those instances where there is evident nonuniformity in the values of the function u within the neighborhood of a vertex v one may need to further control the influence of the edges with large increases. In those cases we may need to use the expression (28) at the bottom of the page as the direction of the gradient.

Under (28) the gradient is given as a weighted sum of the direction of all adjacent edges to a node. The weights are appropriately chosen so that edges along which the change in the value of the function is closer to the maximum contribute more to the gradient direction. The parameter p we can tune to determine how strongly such edges affect the final result. Finally, ε is a small constant that guarantees that the denominators in (28) are well defined. In the special case where $\varepsilon = 0$ only the direction of maximum increase survives. For our simulations we have used the empirical values $p = 0.7$ and $\varepsilon = 0.05$. We prove in Section III-F that, when the constant ε vanishes in the limit of large graphs the dominating term in (28) corresponds to the edge with the maximum change in the value of the function establishing that the approximations (28) and (25) asymptotically agree in probability.

The last term that remains to be determined from (21) is the curvature of the contour of each level set of the function u . To this end we first propose an expression for the divergence of a vector valued function on a graph and using this we compute the desired curvature.

C. Curvature Calculation on Graphs

In the continuous case the curvature of a curve given as the contour of a level set of a function u can be computed using

$$\kappa = \text{div} \left(\frac{\nabla u}{\|\nabla u\|} \right). \quad (29)$$

In the preceding sections we derived expressions for the term $\nabla u / \|\nabla u\|$ on a graph. Therefore, the remaining step is to propose an expression for the computation of the divergence of a vector function on a graph. Indeed, let $\mathbf{F} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ be a vector function supported on the nodes of the graph. If we were

$$\frac{\nabla u}{\|\nabla u\|}(v) \triangleq \frac{\sum_{w \in N_{\text{adj}}(v|G)} \text{sign}(u(w) - u(v)) \frac{\mathbf{e}_{vw}}{\left(\max_{s \in N_{\text{adj}}(v|G)} \{ \|u(s) - u(v)\| \} - \|u(w) - u(v)\| \right)^{p+\varepsilon}}}{\left\| \sum_{w \in N_{\text{adj}}(v|G)} \text{sign}(u(w) - u(v)) \frac{\mathbf{e}_{vw}}{\left(\max_{s \in N_{\text{adj}}(v|G)} \{ \|u(s) - u(v)\| \} - \|u(w) - u(v)\| \right)^{p+\varepsilon}} \right\|}. \quad (28)$$

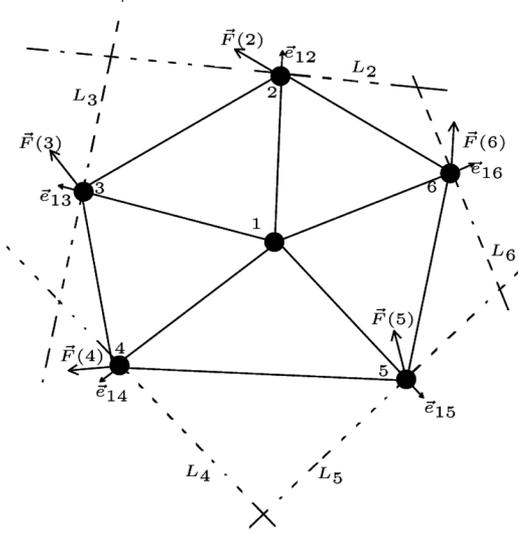


Fig. 8. Computing divergence on a graph: \vec{e}_{uv} denotes the unit vector parallel to the edge (u, v) , L_i the length of the line segment between the perpendicular to the edges lines and $\vec{F}(u)$ the vector value of the function \vec{F} on the node u .

working on the continuous case the standard definition of the *divergence* of \mathbf{F} at a point \mathbf{x} is defined as

$$\operatorname{div}\mathbf{F}(\mathbf{x}) = \lim_{S \rightarrow \{\mathbf{x}\}} \frac{\oint_{\Gamma(S)} \mathbf{F} \cdot \mathbf{n} dl}{|S|},$$

where S is a two dimensional region, $\Gamma(S)$ its boundary, \mathbf{n} the outward unit normal to that boundary, and $|S|$ its enclosed area.

An abstraction of the standard definition of divergence on a graph using only local information at node v can be considered as in Fig. 8. In this context, $\Gamma(S)$ is the polygon defined by the lines perpendicular to the edges adjacent to the node and S is the area enclosed within. The line integral is approximated by the Riemann sum of the inner products. The latter is taken between the value of the vector field \mathbf{F} on each node and the outward unit normal to the boundary, i.e., the unit vector in the direction of each edge.

Concluding, the divergence of \mathbf{F} on a graph can be approximated by

$$\operatorname{div}\mathbf{F}(v) = \frac{\sum_{w \in N_{\text{adj}}(v|G)} L_w \mathbf{F}(w) \cdot \mathbf{e}_{vw}}{S(v)} \quad (30)$$

where

- L_w corresponds to the length of the perpendicular line segment to the edge \mathbf{e}_{vw} . The segment is defined by the intersection between successive lines as illustrated in Fig. 8, and
- $S(v)$ corresponds to the area of the polygon between the perpendicular to the edges lines.

Summarizing, in order to compute the curvature of the contour of the level set of a graylevel function u on a graph we first calculate the normalized gradient vector of the embedding real function on each node as described in Section III-B. Then, the curvature can be calculated using (29). At this point we can perform all the necessary calculations to compute the curvature of the contour of the level set of a graylevel function u on a graph. To illustrate the behavior of the expression proposed consider a circular shaped contour as in Fig. 9. We would expect the curvature for all points on the circle to be a positive number, if we

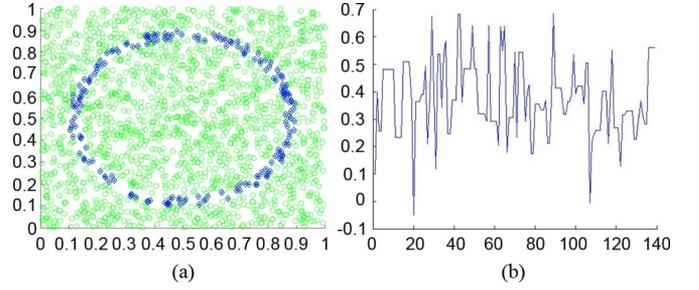


Fig. 9. The curvature on a circle calculated with the method proposed in Section III-C. We omit the edges for illustration purposes (a) Circle on geometric graph (b) The curvature on the circle.

were working in the continuous setting. On a graph, the curvature cannot be expected to be constant but the average value should be positive and the curvature at each point should oscillate around the average. This behavior is captured in Fig. 9.

D. Last Steps

The external image-dependent force is given by the edge-stopping function $g(I)$. The main goal of $g(I)$ is actually to stop the evolving curve when it arrives to the objects boundaries. Among the many choices proposed in the literature, we use the following taken from [9], [25]:

$$g(\|\nabla I_\sigma\|) = \frac{1}{1 + \frac{\|\nabla I_\sigma\|^2}{\lambda^2}} \quad (31)$$

where $I_\sigma = I * G_\sigma$ is the image convolution with a 2D isotropic Gaussian G_σ of variance σ^2 .

The computation of the smoothed version I_σ of I involves the definition of a convolution operation on graphs. Let (V, E) denote the underlying graph. For each $v, w \in V$ define the function $G_\sigma^w(v)$ by

$$G_\sigma^w(v) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{d(v, w)^2}{2\sigma^2}\right). \quad (32)$$

where $d(v, w)$ is the Euclidean distance between vertices. The smoothed version I_σ can be computed by mimicking the continuous convolution operation as follows:

$$I_\sigma(v) = \sum_{w \in V} I(w) G_\sigma^w(v). \quad (33)$$

The above is a natural interpretation of standard convolution on graphs.

E. Active Contours on Graphs—Algorithm 1

Here we combine all the previous approximations to the PDE for geodesic active contours and summarize the algorithm for automatic graph segmentation.

Consider a graph $G = (V, E)$ and let a function $I : V \rightarrow \mathbb{R}$ assign a real value to each of the graph nodes.

Algorithm-Active Contour on Graphs:

- 1) Compute the smoothed version I_σ of I as described in Section III-D.
- 2) Compute the magnitude of ∇I_σ as described in Section III-A and then compute the function $g(\|\nabla I_\sigma\|)$.

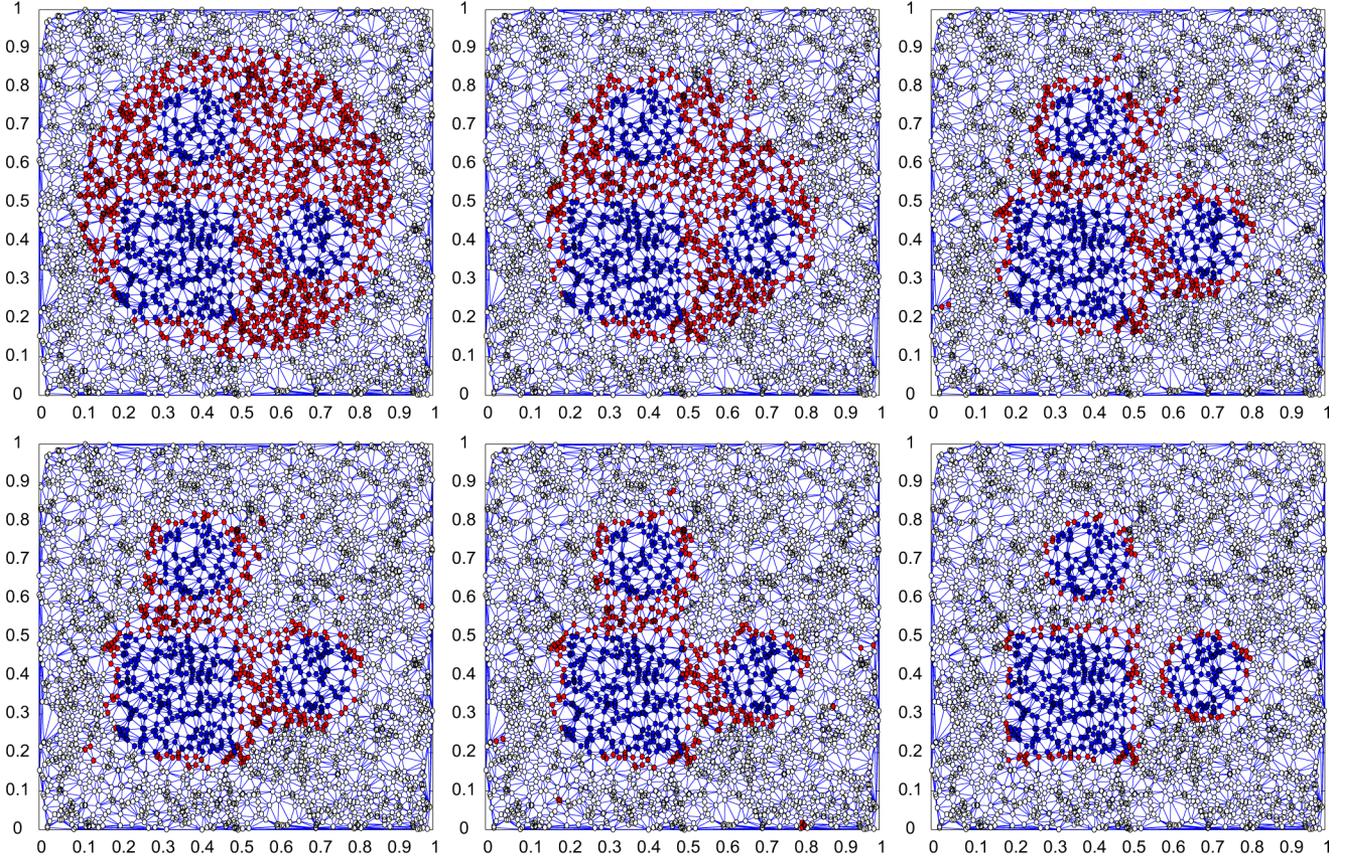


Fig. 10. Illustration of the active contour algorithm on graphs for finding three distinct objects on a graph. Note the change in the contour's topology. Time evolves from top left to bottom right. For this example, $N = 4000$, $\rho = 0.04$. The items in blue are the objects to be found. The image in this case takes the value 1 within the objects and zero in the background.

- 3) Initiate the algorithm with a set of nodes that contains the objects to be found and let ϕ_o denote the signed distance function from the contour of the determined set.
- 4) For each $r \in \mathbb{N}$ compute $\nabla\phi_{r-1}$, $\|\nabla\phi_{r-1}\|$ and the curvature κ at each node v as described in Sections III-A and III-C. Iterate according to the following difference equation:

$$\phi_{r+1} - \phi_r = g(\|\nabla I_\sigma\|) \|\nabla\phi_{r-1}\| (c + \kappa) + g(\|\nabla I_\sigma\|) \cdot \nabla\phi_{r-1}, \quad c \geq 0.$$

Fig. 10 illustrates the algorithm-1 for the case of finding the boundaries of three distinct objects.

F. Convergence in Probability

In this section we prove the convergence of our approximations to the true value of the operators at each point in the limit of large number of nodes for a specific type of graphs, the geometric random graphs.

A *geometric random graph* is a random undirected graph drawn on a bounded region, e.g., the unit square, and is denoted by $G(n, \rho(n))$. The graph $G(n, \rho(n))$ consists of n vertices placed at random uniformly and independently on the region and a set of edges E connecting two vertices u, v if and only if the distance between them is at most $\rho(n)$, i.e., $d(u, v) \leq \rho(n)$.

First, we prove that the simplest approximation of the gradient converges to the true value of the operator at every point in probability as the number of nodes grows large.

Theorem 1: Let $u : [0, 1]^2 \rightarrow \mathbb{R}$ be a twice differentiable function on the unit square and let $G(n, \rho(n))$ be a geometric random graph embedded in the same domain. Let v be a node of $G(n, \rho(n))$ and let N_v denote the set of nodes adjacent to v . Then, if $\rho(n) = 1/n^\alpha$ where $\alpha < 1/2$,

$$\lim_{n \rightarrow \infty} \max_{w \in N_v} \left\{ \frac{u(w) - u(v)}{d(w, v)} \right\} = |\nabla u(v)|,$$

and

$$\lim_{n \rightarrow \infty} \mathbf{e}_{vw(v)} = \frac{\nabla u(v)}{|\nabla u(v)|},$$

in probability, where $w(v) = \operatorname{argmax}_{w \in N_v} \{u(w) - u(v)/d(w, v)\}$.

Proof: The second implication directly implies the first, therefore we will just prove the second. To this end, let $\hat{\theta} = \angle \mathbf{e}_{vw(v)}$ denote the angle of the random unit vector pointing towards the direction of the largest increase within the node v 's neighborhood and let $\theta = \angle \nabla u(v)$. Note that while θ is deterministic, $\hat{\theta}$ is a random variable whose convergence we wish to establish. Consider some $\varepsilon > 0$. In order to prove convergence in probability, we need to bound the probability of a deviation of $\hat{\theta}$ away from θ , namely $\mathbb{P}(|\hat{\theta} - \theta| > \varepsilon)$.

For $|\hat{\theta} - \theta| > \varepsilon$ it has to be the case that there exists some neighbor w of v for which $|\angle \mathbf{e}_{vw} - \theta| > \varepsilon$ and $\Delta_{vw}(u) >$

$\Delta_{vz}(u)$ for all z such that $|\angle \mathbf{e}_{vz} - \theta| < \varepsilon$, where $\Delta_{vw}(u) = u(v) - u(w)/d(v, w)$, or the set $\mathcal{Z} = \{z : |\angle \mathbf{e}_{vz} - \theta| < \varepsilon\}$ is empty.

In other words,

$$\begin{aligned} & \mathbb{P}(|\hat{\theta} - \theta| > \varepsilon) \\ & \leq \mathbb{P}(\cup_{w \in N_v, |\angle \mathbf{e}_{vw} - \theta| > \varepsilon} \Delta_{vw}(u) > \max_{z \in \mathcal{Z}} \Delta_{vz}(u) \mid \mathcal{Z} \neq \emptyset) \\ & \quad + \mathbb{P}(\mathcal{Z} = \emptyset). \end{aligned}$$

We proceed by showing that each of the terms on the right hand side of the above equation converges to zero. To this end, note that, the size $|\mathcal{Z}|$ of \mathcal{Z} is a Bernoulli random variable and therefore

$$\mathbb{P}(\mathcal{Z} = \emptyset) = \mathbb{P}(|\mathcal{Z}| = 0) = \left(1 - \frac{2\pi\varepsilon}{n^{2\alpha}}\right)^n,$$

which converges to zero if $\alpha < 1/2$ using the fact that

$$\left(1 - \frac{2\pi\varepsilon}{n^{2\alpha}}\right)^{n^{2\alpha}} \rightarrow e^{-1}.$$

Moreover, by the Taylor expansion,

$$\Delta_{vw}(u) = D_w u(v) + \frac{O(d(w, v))}{d(w, v)}$$

where $D_w u(v)$ is the directional derivative of u in the direction of \mathbf{e}_{vw} . From the definition of gradient we get

$$D_w u(v) = |\nabla u(v)| \cos(\phi_w)$$

where $\phi_w = \theta - \angle \mathbf{e}_{vw}$.

Summarizing the above, for any $w \in N_v$, $|\angle \mathbf{e}_{vw} - \theta| > \varepsilon$, and any $z \in \mathcal{Z}$

$$\begin{aligned} \mathbb{P}(\Delta_{vw} > \Delta_{vz}(u)) &= \mathbb{P}\left(|\nabla u(v)| \cos(\phi_w) + \frac{O(d(w, v))}{d(w, v)} > \right. \\ & \quad \left. + |\nabla u(v)| \cos(\phi_z) + \frac{O(d(z, v))}{d(z, v)}\right). \end{aligned}$$

When n is large we get that

$$\begin{aligned} \lim_{n \rightarrow \infty} d(w, v) &< \lim_{n \rightarrow \infty} \rho(n) = 0, \\ &\text{and} \\ \lim_{n \rightarrow \infty} d(z, v) &< \lim_{n \rightarrow \infty} \rho(n) = 0. \end{aligned}$$

Therefore, for any $w \in N_v$, $|\angle \mathbf{e}_{vw} - \theta| > \varepsilon$, and any $z \in \mathcal{Z}$

$$\begin{aligned} \lim_{n \rightarrow \infty} \mathbb{P}(\Delta_{vw} > \Delta_{vz}(u)) \\ = \lim_{n \rightarrow \infty} \mathbb{P}(\cos(\phi_w) > \cos(\phi_z)) &= 0 \end{aligned}$$

concluding the proof. \blacksquare

The next step is to prove convergence for the normalized smoothed approximation proposed by (34) at the bottom of the page.

The main idea of the proof is to show that this approximation in probability approaches the one of Theorem 1 thus establishing convergence in probability.

Theorem 2: Let $u : [0, 1]^2 \rightarrow \mathbb{R}$ be a twice differentiable function on the unit square and let $G(n, \rho(n))$ be a geometric random graph embedded in the same domain. Let v be a node of $G(n, \rho(n))$ and let N_v denote the set of nodes adjacent to v . Then, if $\rho(n) = 1/n^\alpha$ where $\alpha < 1/2$ and $\varepsilon(n) \rightarrow 0$, the approximation of (34) for the direction of the gradient converges in probability to its true value.

Proof: Let $\bar{\theta}$ be the random variable denoting the angle of the approximation and let θ and $\hat{\theta}$ as defined in the proof of Theorem 1. By the triangular inequality we get that, for all $\varepsilon > 0$,

$$\mathbb{P}(|\bar{\theta} - \theta| < \varepsilon) \geq \mathbb{P}\left(|\hat{\theta} - \theta| < \frac{\varepsilon}{2}\right) + \mathbb{P}\left(|\hat{\theta} - \bar{\theta}| < \frac{\varepsilon}{2}\right).$$

Therefore, in order to establish the result we need to show that for all $\varepsilon > 0$

$$\lim_{n \rightarrow \infty} \mathbb{P}\left(|\hat{\theta} - \theta| < \frac{\varepsilon}{2}\right) = 0.$$

Let \mathbf{e}_ϕ denote the unit vector in the direction of $\phi \in [0, 2\pi)$. By the used approximation, $\mathbf{e}_{\bar{\theta}}$ is given as the weighted sum of unit vectors in the direction of all the edges adjacent to node v , i.e.,

$$\mathbf{e}_{\bar{\theta}} = w_{\hat{\theta}} \mathbf{e}_{\hat{\theta}} + \sum_{w \in N_v(G), \mathbf{e}_{vw} \neq \mathbf{e}_{\hat{\theta}}} w_{vw} \mathbf{e}_{vw},$$

Note that, since $\varepsilon(n) \rightarrow 0$, $w_{\hat{\theta}} \rightarrow 1$ and $w_{vw} \rightarrow 0$ for all $w \in N_v(G)$ such that $\mathbf{e}_{vw} \neq \mathbf{e}_{\hat{\theta}}$, the first term only survives in the limit of large n , concluding the proof. \blacksquare

In the next section we present an alternative way to implement geodesic active contours, through a direct minimization of the energy functional associated with each contour. Novel methods combining ideas from energy minimization via graph cuts and classical segmentation techniques like watershed on the regular grid have been lately developed, [14].

IV. ACTIVE CONTOURS VIA GEOCUTS

A. Integral Geometry and Crofton Formulas

In this section we present a formula that relates the length of a curve in \mathbb{R}^2 to a measure of a set of lines intersecting it. The first step will be to describe a reasonable way of assigning a measure to a given subset of straight lines. \blacksquare

$$\frac{|\nabla u|}{\|\nabla u\|}(v) \triangleq \frac{\sum_{w \in N_{\mathcal{E}}(v|G)} \text{sign}(u(w) - u(v)) \frac{\mathbf{e}_{vw}}{\left(\max_{s \in N_{\mathcal{E}}(v|G)} \left\{ \frac{|u(s) - u(v)|}{d(s, v)} \right\} - \frac{|u(w) - u(v)|}{d(w, v)}\right)^p + \varepsilon(n)}}{\left\| \sum_{w \in N_{\mathcal{E}}(v|G)} \text{sign}(u(w) - u(v)) \frac{\mathbf{e}_{vw}}{\left(\max_{s \in N_{\mathcal{E}}(v|G)} \left\{ \frac{|u(s) - u(v)|}{d(s, v)} \right\} - \frac{|u(w) - u(v)|}{d(w, v)}\right)^p + \varepsilon(n)} \right\|}. \quad (34)$$

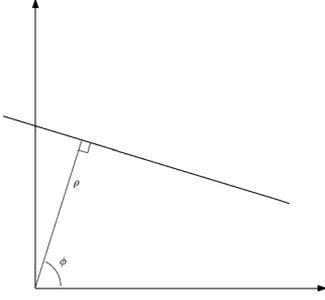


Fig. 11. The parameters ρ and ϕ are the polar coordinates of the foot of the perpendicular from the origin onto the line.

Every straight line on the plane can be parametrized by the distance ρ from the origin and the angle ϕ between the line and the y -axis as illustrated in Fig. 11. Let $\mathcal{L} = \{(\rho, \phi) : \rho \geq 0, \phi \in [0, 2\pi]\}$ be the set of all straight lines on the plane. The Lebesgue measure on this set is defined by the density $d\mathcal{L} = d\rho \cdot d\phi$ and the Lebesgue measure of a subset of straight lines $\mathcal{X} \subset \mathcal{L}$ is given by the integral $\int_{\mathcal{X}} d\mathcal{L}$.

Given the above measure on the space of straight lines, the following Cauchy-Crofton formula connects the Euclidean length $|C|_E$ of a curve C in \mathbb{R}^2 and the measure of the lines intersecting it:

$$|C|_E = \frac{1}{2} \int N_C(L) d\mathcal{L} \quad (35)$$

where $N_C(L)$ corresponds to the number of times any given line L intersects C .

Similarly, if we consider a 2D Riemannian metric space with a metric $D(\cdot)$, the Riemannian length of contour C is equal to

$$|C|_R = \frac{1}{2} \int \frac{\det(D)}{(\mathbf{u}_L^T D \mathbf{u}_L)^{3/2}} N_C(L) d\mathcal{L} \quad (36)$$

where \mathbf{u}_L is the unit vector in the direction of the line L . For more details on the above the reader may consult [2].

B. Graph Cut Methods

Consider an undirected graph $G = (V, E)$, defined as a set of nodes V and edges E connecting these nodes. We assign to each edge in the graph a nonnegative weight cost w_e .

We define also two problem specific nodes called terminals. A cut is then a subset of edges whose removal separates the terminals. The cost associated with each cut C is defined as the sum of the costs of the edges that it contains.

$$|C| = \sum_{e \in C} w_e.$$

Finding the globally minimum cut on a graph with two terminals is a well known problem and it can be efficiently computed through standard max-flow or combinatorial techniques (e.g., [5], [13], [15]).

Therefore it is natural for any cut $C \subset E$ on a graph G to be associated to the length $|C|_G = \sum_{e \in C} w_e$. The connection between the *cut metric* defined and the length of a curve on a Riemannian space is now offered by the Cauchy-Crofton equations that we presented in Section IV-A. This idea has been analyzed

in [4] for the canonical grid and we extend it to graphs in the following section.

C. Euclidean Cut Metric on Planar Graphs

We start by summarizing the basic results from [4] for comparison purposes. Neighborhood systems of a regular grid can be described by a set $\{e_i\}_{i=1}^n$ of distinct vectors, enumerated in increasing order of angular orientation. Each such vector generates a family of edge lines on the grid and the distance between the nearest lines in the family is

$$\Delta\rho_k = \frac{\delta^2}{|e_k|}$$

where δ is the cell size of the grid.

Consider now a contour C in the same plane as the regular grid graph. C being a closed curve corresponds to a cut on G . Then, if each edge belonging to the family of the vector e_k is assigned a weight w_k the cut-metric of the cut defined by C is given by

$$|C|_G = \sum_{e \in C} w_e = \sum_k N_C(k) w_k,$$

where $N_C(k)$ is the number of intersections of the contour C with lines from the family associated with vector e_k . On the other hand, we get from the Cauchy-Crofton formula that the Euclidean length of the contour is

$$|C|_E = \frac{1}{2} \int N_C(L) d\mathcal{L} \quad (37)$$

$$\approx \sum_k N_C(k) \frac{\delta^2 (\phi_{k+1} - \phi_k)}{2|e_k|}. \quad (38)$$

Therefore by choosing constant edge weights within each family of edge lines as

$$w_k = \frac{\delta^2 \cdot \Delta\phi_k}{2 \cdot |e_k|} \quad (39)$$

we get

$$|C|_G \approx |C|_E.$$

In our case, there does not exist a basis of vectors that uniformly represent the edges of an arbitrary graph. Therefore the analysis will change. Our goal is to assign a weight w_e on each edge $e \in E$ of our graph so that the cut metric of a contour is a good approximation of the Euclidean metric. To this end if we write the Cauchy-Crofton formula as

$$\begin{aligned} |C|_E &= \frac{1}{2} \int N_C(L) d\mathcal{L} \\ &= \int_{(\rho, \phi) \in [0, +\infty) \times [0, \pi]} N_C(\rho, \phi) d\rho d\phi \end{aligned}$$

and the cut metric as

$$|C|_G = \sum_{e \in C} w_e,$$

we can conclude that in order for $|C|_E \approx |C|_G$ the weight w_e should be an approximation of the Lebesgue measure of the area in the (ρ, ϕ) plane that edge e represents. This area is the

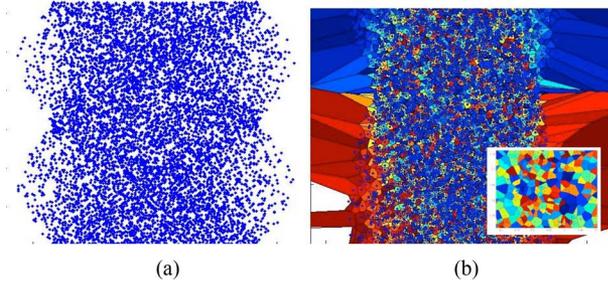


Fig. 12. Calculation of the edge weights for the Euclidean cut metric (a) The (ρ, ϕ) space of the edges of a geometric random graph (b) The Voronoi diagram of the space of edges. The weight of each edge is equal to the area of the corresponding cell.

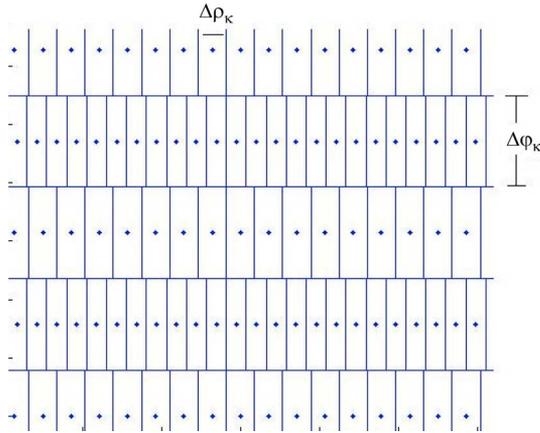


Fig. 13. The Voronoi diagram of the space of edges of a regular grid.

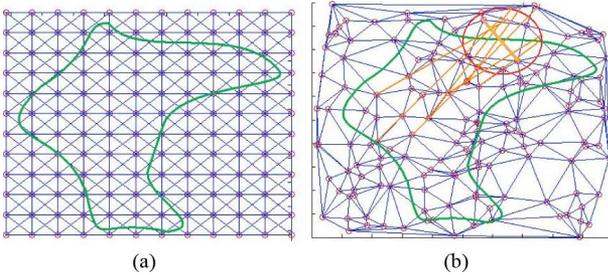


Fig. 14. Weight modification (a) For the case of a regular grid the contour at each point can be intersected by just one member of the family (b) For the case of an arbitrary graph a contour that is intersected by an edge can also be intersected by the extension of the lines defined from other edges in the graph.

corresponding Voronoi cell of the point (ρ_e, ϕ_e) as illustrated in Fig. 12 and its area is denoted by Λ_e .

As a sanity check we observe in Fig. 13 that in the case of a regular 2D grid the area of the cell of each edge is proportional to the weight assigned to it by (39) in [4].

Finally another important difference between the regular grid and the arbitrary graph is the phenomenon depicted in Fig. 14. Whenever a contour crosses an edge then it also crosses the lines defined by other edges in the graph. This can be taken into account by modifying the edge weight and defining it as the sum of the areas associated to all the lines that cross each edge. This is a computationally expensive process but increases the precision of the approximations since more intersections are taken into account. This calculation can be performed offline, just using the graph information and therefore the computational burden does

not transfer to the execution of the max flow problem. Specifically, for each pair of edges we calculate their point of intersection and add the (ρ, ϕ) area of one to the weight of the other if and only if this point lies within the adjacent nodes defining the edge. This modification, is used in order to increase the accuracy of the Cauchy-Crofton approximation and is not necessary for the algorithm. Moreover it implicitly assumes that all the lines that are crossing a specific edge, are also crossing a curve which is cut by that edge. This is an approximation in the case of finite graphs but the error diminishes in the limit of large, dense graphs.

Summarizing, let $e = (p, q)$ be the edge connecting the nodes p and q and let

$$Q_e = \{e' \in E : \text{the straight lines defined by } e \text{ and } e' \text{ cross between } p \text{ and } q\}.$$

Then if we assign to each edge a weight

$$w_e = \frac{1}{2} \left(\sum_{j \in N_e} \Lambda_j + \Lambda_e \right), \quad (40)$$

the cut metric and the Euclidean metric are approximately the same $|C|_G \approx |C|_E$.

Similar to the previous section we can directly generalize the above to Riemannian spaces using (36). In that case the corresponding weights are given by

$$w_e = \frac{1}{2} \sum_{z \in Q_e \cup \{e\}} \Lambda_z \frac{\det(D(p_z))}{(\mathbf{z}^T D(p_z) \mathbf{z})^{3/2}}, \quad (41)$$

where p_z is a node from which edge z is originating.

D. Active Contours on Graphs-Algorithm 2

Summarizing, using the weight approximation from (40) we can complete all the necessary steps to implement geodesic active contours using GeoCuts. Specifically, consider a graph $G = (V, E)$ and let a function $I : V \rightarrow \mathbb{R}$ assign a real value to each of the graph nodes.

- 1) Let $D(p) = g(|\nabla I(p)|)\mathbf{I}$, where g is defined in Section III-D and \mathbf{I} is the 2×2 identity matrix.
- 2) Assign to each edge a weight using (41).
- 3) Define a terminal nodes at the boundary and inside each object of interest.
- 4) Solve the minimum cut problem by using combinatorial optimization tools (e.g., [5], [13], [15]).

An example of the execution of our algorithm-2 is given in Fig. 15. We have solved the minimum cut problem using a gradient descent type algorithm that allows us to observe the deformation of an initial contour. For the Min-Cut problem though, state of the art algorithms have been developed that computationally outperform the gradient descent method that we chose and it is an interesting research direction to implement and optimize the algorithm for the solution of the combinatorial problem posed.

V. CONCLUSIONS AND DISCUSSION

In this paper our contribution is twofold. First, we propose an approximation to level set implementation of morphological

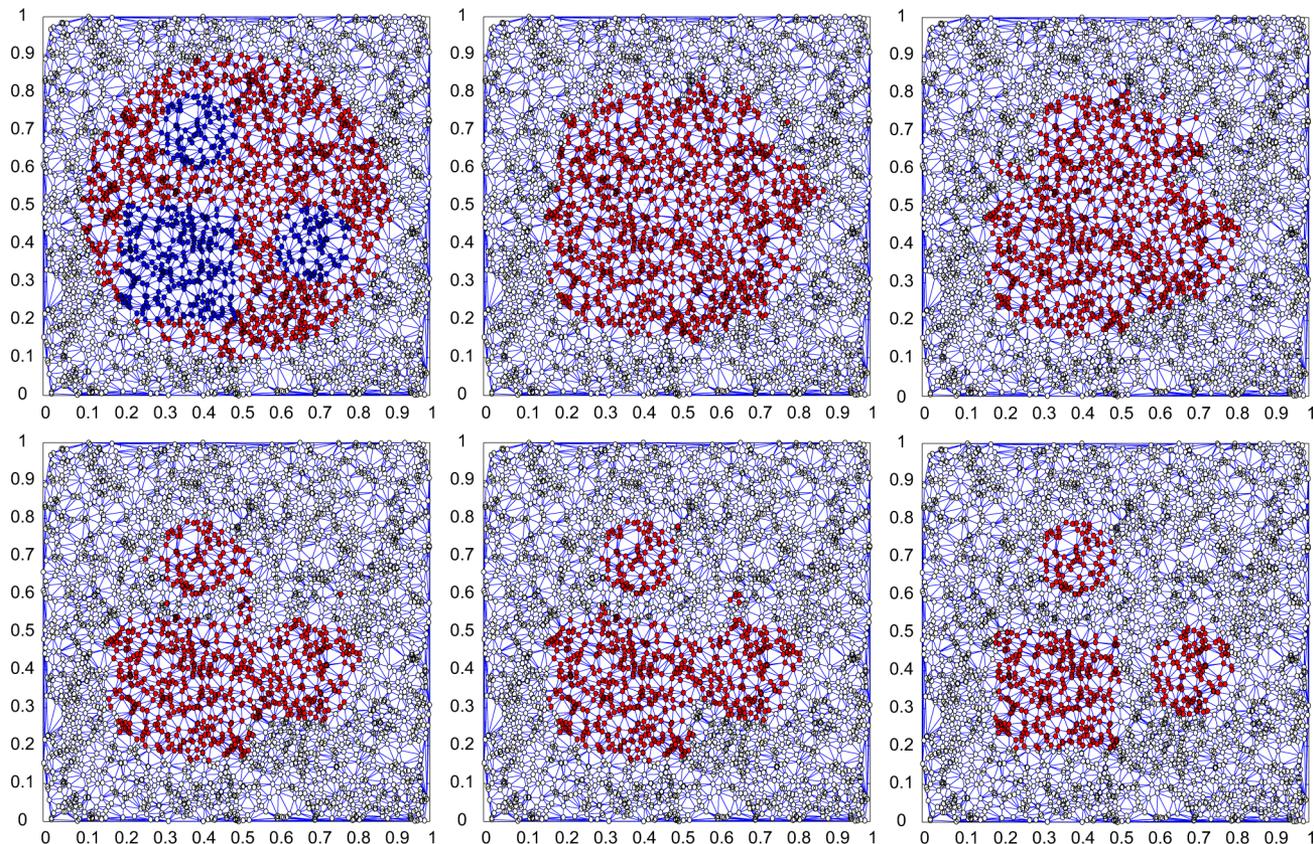


Fig. 15. Illustration of the active contour algorithm using graph cuts on graphs for finding three distinct objects on a graph. Time evolves from top left to bottom right.

operators and geodesic active contours on arbitrary graphs. Our motivation comes from the importance and the success of such concepts and techniques in image analysis as well as the existence of a strong theoretical background on graph morphology. Based on these concepts and using heuristic approximations to operators from vector calculus we propose a difference equation approximation for the evolution of geodesic active contours on graphs.

Second, we extend existing work on implementation of active contours using graphcuts on the regular grid to arbitrary graphs. The two methods are very different in the sense that the first approximates the solution to the continuous space minimization problem while the second approximates the minimization problem itself and transforms it to a min-cut problem. We believe that both methods illuminate interesting research directions toward theoretical guarantees for the heuristic approaches. Finally, we note that the execution time for our first method is much smaller than our gradient-descent approach to the second method. On the other hand combinatorial optimization is a fast evolving field and results may be much different using the state of the art Min-Cut algorithms.

A contribution of our paper towards future potential applications is the illustration of those techniques for automatic object detection and segmentation on graphs. The methods illustrated involve the use of heuristic approximations, which, on top of the intrinsic noise of the geodesic active contour methods, involve additional noise due to the graph approximations and may lead

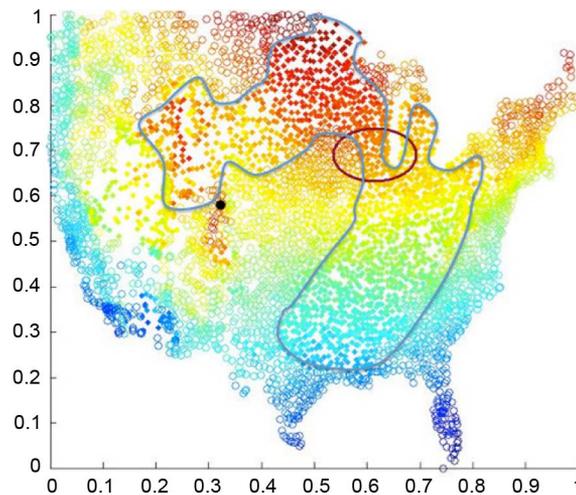


Fig. 16. The graph nodes (circles and diamonds) represent rainfall data for each city of USA. Segmentation via our graph GAC identifies essentially two regions (indicated by diamonds) and connected through the narrow neck indicated by the ellipsis): one corresponding to the Humid Continental Dfb climate zone and one to the Humid continental Dfa and Subtropical climate zone.

to instability in specific settings. Therefore, exploring the robustness and the practical convergence of the aforementioned techniques as well as improving the proposed approximations towards faster and better convergence is an open problem and topic of future research. A first promising step in the former direction is our proof of the convergence in probability of the

discretized gradient operators we used for GACs on geometric random graphs with large number of nodes.

An example of the application of our techniques on geographical information systems is illustrated in Fig. 16. In this application, we collected the average yearly rainfall values for 4000 cities in the USA. These values define the value of the grayscale function defined on each node. Assuming correlation between nodes that are geographically close to each other we produce a graph according to the following process: if the distance between two cities is smaller than a threshold then we assume that an edge that connects them exists. We produce the segmentation result using the geodesic active contour on graphs algorithm and it identifies two regions with similar rainfall yearly statistics. Actually these regions correspond to different climate zones of the USA. There is a thin neck connection between these two regions. Overall, the curve evolves towards the region that differentiates from its background and stabilizes close to areas with very characteristic meteorological behavior.

In general, our methods, can be applied to various areas including, image processing, social networks, geographical systems, etc. We foresee strong applicability in graph-based image processing and computer vision tasks where the image information is sampled sparsely and irregularly. Moreover, automatic community detection and automatic identification of groups with similar behavior with respect to a specific characteristic is a topic that has lately received great attention in the scientific community, given the large amount of data that is produced by the trending social networks.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for providing useful references and constructive comments which helped improve this paper. They also thank Epameinondas Antonakos at NTUA for providing Fig. 1.

REFERENCES

[1] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing," *Archiv. Rat. Mech.*, vol. 123, no. 2, pp. 199–257, Jun. 1993.

[2] A. Baddeley and E. B. Vedel-Jensen, *Stereology for Statisticians*. Boca Raton, FL: Chapman & Hall/CRC, 2004.

[3] A. Bensoussan and J. Menaldi, "Difference equations on weighted graphs," *J. Convex Anal.*, vol. 12, no. 1, pp. 13–44, 2005.

[4] Y. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *Proc. Int. Conf. Comput. Vis.*, 2003, pp. 26–33.

[5] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[6] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[7] M. Breuß and J. Weickert, "Highly accurate schemes for pde-based morphology with general convex structuring elements," *Int. J. Comp. Vis.*, vol. 92, no. 2, pp. 132–145, 2011.

[8] R. W. Brockett and P. Maragos, "Evolution equations for continuous-scale morphological filtering," *IEEE Trans. Signal Process.*, vol. 42, no. 12, pp. 3377–3386, Dec. 1994.

[9] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerische Mathematik*, vol. 66, pp. 1–31, 1993.

[10] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.

[11] A. Chambolle and J. Darbon, "On total variation minimization and surface evolution using parametric maximum flows," *Int. J. Comput. Vis.*, vol. 84, no. 3, pp. 288–307, 2009.

[12] S.-Y. Chung and C. A. Berenstein, "Omega-harmonic functions and inverse conductivity problems on networks," *SIAM J. Appl. Math.*, vol. 65, pp. 1200–1226, 2005.

[13] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*. New York: Wiley, 1998.

[14] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watershed: A unifying graph-based optimization framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1384–1399, Jul. 2011.

[15] C. Couprie, L. Grady, H. Talbot, and L. Najman, "Combinatorial continuous maximum flow," *SIAM J. Img. Sci.*, vol. 4, no. 3, pp. 905–930, Sep. 2011.

[16] J. Cousty, L. Najman, and J. Serra, "Some morphological operators in graph spaces," in *Proc. Int. Symp. Math. Morphol., LNCS 5720*, 2009.

[17] L. J. Grady and J. R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. New York: Springer, 2010.

[18] H. J. A. M. Heijmans, *Morphological Image Operators*. Boston, MA: Academic, 1994.

[19] H. J. A. M. Heijmans, P. Nacken, A. Toet, and L. Vincent, "Graph morphology," *J. Vis. Commun. Image Represent.*, vol. 3, no. 1, pp. 24–38, 1992.

[20] A. N. Hirani, "Discrete exterior calculus," Ph.D. dissertation, California Inst. of Technol., Pasadena, 2003.

[21] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[22] B. Kimia, A. Tannenbaum, and S. Zucker, "On the evolution of curves via a function of curvature. I. The classical case," *J. Math. Anal. Appl.*, vol. 163, pp. 438–458, 1992.

[23] N. Komodakis, N. Paragios, and G. Tziritas, "MRF energy minimization and beyond via dual decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 531–552, Mar. 2011.

[24] *Image Processing and Analysis With Graphs: Theory and Practice*, O. Lezoray and L. J. Grady, Eds. Boca Raton, FL: CRC, 2012.

[25] R. Malladi, J. A. Sethian, and B. C. Vemuri, "A fast level set based algorithm for topology-independent shape modeling," *J. Math. Imag. Vis.*, vol. 6, pp. 269–289, 1996.

[26] P. Maragos, "Partial differential equations for morphological scale-spaces and eikonal applications. In," in *The Image and Video Processing Handbook*, A. C. Bovik, Ed., 2nd ed. New York: Elsevier Academic Press, 2005, pp. 587–612.

[27] G. Matheron, *Random Sets and Integral Geometry*. New York: Wiley, 1975.

[28] T. McInerney and D. Terzopoulos, "Topologically adaptable snakes," in *Proc. Int. Conf. Comput. Vis.*, 1995, pp. 840–845.

[29] S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Comp. Phys.*, vol. 79, pp. 12–49, 1988.

[30] G. Sapiro, R. Kimmel, D. Shaked, B. Kimia, and A. Bruckstein, "Implementing continuous-scale morphology via curve evolution," *Pattern Recognit.*, vol. 26, no. 9, pp. 1363–1372, 1993.

[31] "Theoretical Advances," in *Image Analysis and Mathematical Morphology*, J. Serra, Ed. New York: Academic, 1988, vol. 2.

[32] V.-T. Ta, A. Elmoataz, and O. Lézoray, "Nonlocal pdes-based morphology on weighted graphs for image and data processing," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1504–1516, Jun. 2011.

[33] L. Vincent, *Graphs Math. Morphol.*, vol. 16, no. 4, pp. 365–388, 1989.



Kimon Drakopoulos (M'09) received the diploma in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2009 and the M.Sc. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, in 2011.

From 2011 to present, he is a Ph.D. candidate at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA. His current research interests include social network analysis, applied probability, game theory and network economics.



Petros Maragos (F'96) received the diploma in electrical engineering from the National Technical University of Athens (NTUA) in 1980 and the MScEE and PhD degrees from the Georgia Institute of Technology (Georgia Tech), Atlanta, in 1982 and 1985. In 1985, he joined the faculty of the Division of Applied Sciences, Harvard University, Cambridge, Massachusetts, where he worked for eight years as professor of electrical engineering. In 1993, he joined the faculty of the School of Electrical and Computer Engineering at Georgia Tech. During

parts of 1996–1998 he was on sabbatical, working as director of research at the Institute for Language and Speech Processing, Athens. Since 1998, he has been working as a professor at the NTUA School of Electrical and Computer Engineering. His research and teaching interests include signal processing, systems theory, pattern recognition, and their applications to image processing and computer vision, speech processing and recognition, multimedia, and cognitive systems. He has served as an associate editor for the IEEE TRANSACTIONS ON

ACOUSTICS, SPEECH, AND SIGNAL PROCESSING and IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and as an editorial board member for the journals *Signal Processing*, *Visual Communications and Image Representation*, *Mathematical Imaging and Vision*. He has served as cochair or co-organizer of several conferences, including VCIP'92, ISMM'96, VLBV'01, MMSP'07, ECCV'10, EUSIPCO'12, and as a member of three IEEE SPS committees. His research has received several awards, including a 1987 US National Science Foundation Presidential Young Investigator Award, the 1988 IEEE SPS Young Author Paper Award for the paper 'Morphological Filters,' the 1994 IEEE SPS Senior Award and the 1995 IEEE W.R.G. Baker Prize Award for the paper 'Energy Separation in Signal Modulations with Application to Speech Analysis,' the 1996 Pattern Recognition Society's Honorable Mention Award for the paper 'Min-Max Classifiers,' the 2007 EURASIP Technical Achievements Award for contributions to nonlinear signal, image and speech processing, and the best paper award of the IEEE CVPR-2011 gesture workshop. He was elected a fellow of IEEE in 1995 and of EURASIP in 2010.