Book Chapter in Innovations for Shape Analysis: Models and Algotithms,
edited by M. Breuss, A. Bruckstein and P. Maragos, Springer, 2012.

Segmentation and Skeletonization on Arbitrary Graphs Using Multiscale Morphology and Active Contours

Petros Maragos and Kimon Drakopoulos

Abstract In this chapter we focus on formulating and implementing on abstract domains such as arbitrary graphs popular methods and techniques developed for image analysis, in particular multiscale morphology and active contours. To this goal 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's nodes. 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 via segmentation. Finally, using these novel ideas, we propose a method for multiscale shape skeletonization on arbitrary graphs.

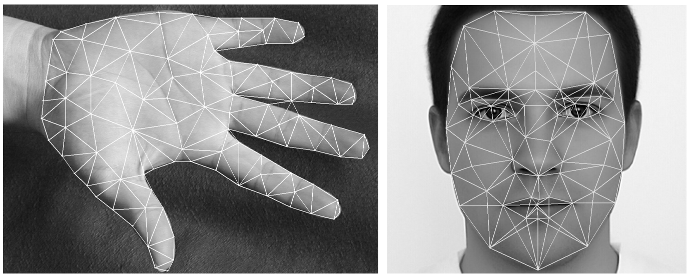
1 Introduction

Graph-theoretic approaches have become commonplace in computer vision. Examples include the graph-cut approaches to segmentation [8, 7, 24, 16] and the statistical inference on discrete-space visual data with graphical models [42]. In most of these cases, the image graphs are regular grids that result from uniform sampling of continuous space. In addition, in nowadays science and technology there exist both low-level and high-level visual data as well as many other types of data defined on *arbitrary graphs* with irregular spacing among their vertices. Examples from the vision area include region-based or part-based object representations, cluster analysis in pattern recognition, and graph-based deformable models for representing and rec-

Petros Maragos
National Technical University of Athens, School of Electrical and Computer Engineering, Athens 15773, Greece. e-mail: maragos@cs.ntua.gr

Kimon Drakopoulos
Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, USA. e-mail: kimondr@mit.edu

ognizing shapes such as faces and gestures [15]. Two such examples from the last area are shown in Fig. 1. Examples from non-vision areas include network problems modeled with graphs, such as social nets, geographical information systems, and communications networks.



(a) Hand on a graph.

(b) Face on a graph.

Fig. 1 Representing image or more general visual information on graphs.

In this chapter we explore theoretically and algorithmically three topics related to shape morphology on arbitrary graphs: multiscale morphology on graphs, geodesic active contours on graphs, and multiscale skeletonization on graphs.

An important part in our work is how to define multiscale morphological operators on arbitrary graphs. We begin to approach this problem algebraically by extending the lattice definitions of morphological operators on arbitrary graphs which have been introduced in [41, 20] with some recent work in [17]. Then we focus on our major approach which is based on discretizing the PDEs generating continuous-scale morphological operators [1, 11] and the PDEs moving geodesic active contours [13] on arbitrary graphs. In this latter direction, a first approach to approximate morphological operators on graphs through mimicking the corresponding PDEs has been studied in Ta et al. [39]. 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 [14, 4] by introducing discrete gradients and Laplacian and by studying Dirichlet and Neumann boundary value problems on graphs.) In the rest of our work, we propose approximations for computing the differential terms required in applying the technique of geodesic active contours to object detection on graphs. Finally, the modeling of multiscale morphology on graphs allows us to develop a method for multiscale skeletonization of shapes on arbitrary graphs.

2 Multiscale Morphology on Graphs

In this section we first review some basic concepts from lattice-based morphology. Then, we focus our review on 1) multiscale morphological image operators on a Euclidean domain, either defined algebraically or generated by nonlinear PDEs, and 2) on defining morphological operators on arbitrary graphs. Finally, we connect these two areas and define multiscale morphological operators on graphs.

2.1 Background on Lattice and Multiscale Morphology

A general formalization [37, 19] of morphological operators views them as operators on complete lattices. A *complete lattice* is a set $\mathscr{L}$ equipped with a partial ordering $\leq$ such that $(\mathscr{L}, \leq)$ has the algebraic structure of a *partially ordered set* where the supremum and infimum of any of its subsets exist in $\mathscr{L}$. For any subset $\mathscr{K} \subseteq \mathscr{L}$, its *supremum* $\bigvee \mathscr{K}$ and *infimum* $\bigwedge \mathscr{K}$ are defined as the lowest (with respect to $\leq$) upper bound and greatest lower bound of $\mathscr{K}$, respectively. The two main examples of complete lattices used respectively in morphological shape and image analysis are: (i) the power set $\mathscr{P}(E) = \{X : X \subseteq E\}$ 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 $f : E \to \mathscr{T}$ where $\mathscr{T}$ is a continuous or quantized sublattice of $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ and the $\bigvee / \bigwedge$ lattice operations are the supremum/infimum of sets of real numbers. An operator ψ on $\mathscr{L}$ is called **increasing** if $f \leq g$ implies $\psi(f) \leq \psi(g)$. Increasing operators are of great importance; among them four fundamental examples are:

$$\delta \text{ is } \textbf{dilation} \Longleftrightarrow \delta(\bigvee_{i \in I} f_i) = \bigvee_{i \in I} \delta(f_i) \tag{1}$$

$$\varepsilon \text{ is } \textbf{erosion} \Longleftrightarrow \varepsilon(\bigwedge_{i \in I} f_i) = \bigwedge_{i \in I} \varepsilon(f_i) \tag{2}$$

$$\alpha \text{ is } \textbf{opening} \Longleftrightarrow \alpha \text{ is increasing, idempotent, and anti-extensive} \tag{3}$$

$$\beta \text{ is } \textbf{closing} \Longleftrightarrow \beta \text{ is increasing, idempotent, and extensive} \tag{4}$$

where I is an arbitrary index set, *idempotence* of an operator ψ means that $\psi^2 = \psi$, and *anti-extensivity* and *extensivity* of operators α and β means that $\alpha(f) \leq f \leq \beta(f)$ for all f. Operator products mean composition: $\phi\psi(f) = \phi(\psi(f))$. The notation ψ^r means r-fold composition.
Dilations and erosions come in pairs (δ, ε) called **adjunctions** if

$$\delta(f) \leq g \Longleftrightarrow f \leq \varepsilon(g) \tag{5}$$

Such pairs are useful for constructing openings $\alpha = \delta\varepsilon$ and closings $\beta = \varepsilon\delta$. The above definitions allow broad classes of signal operators to be studied under the unifying lattice framework.

In *Euclidean morphology*, the domain E becomes the d-dimensional Euclidean space $\mathbb{E}^d$ where $\mathbb{E}=\mathbb{R}$ or $\mathbb{E}=\mathbb{Z}$. In this case, the most well-known morphological operators are the translation-invariant Minkowski dilations $\oplus$, erosions $\ominus$, openings $\circ$, and closings $\bullet$, which are simple special cases of their lattice counterparts. If $X_{+b}=\{x+b : x\in X\}$ denotes the translation of a set/shape $X\subseteq\mathbb{E}^d$ by $b\in\mathbb{E}^d$, the simple Minkowski set operators are $X\oplus B=\bigcup_{b\in B}X_{+b}$, $X\ominus B=\bigcap_{b\in B}X_{-b}$, and $X\circ B=(X\ominus B)\oplus B$. The set B usually has a simple shape and small size, in which case it is called a *structuring element*. By denoting with $rB=\{rb : b\in B\}$ the r-scaled homothetic of B, where $r\geq 0$, we can define *multiscale* translation-invariant morphological set operators on $\mathbb{R}^d$:

$$\delta_{rB}(X)\triangleq X\oplus rB,\quad \varepsilon_{rB}(X)\triangleq X\ominus rB,\quad \alpha_{rB}(X)\triangleq X\circ rB \tag{6}$$

Similarly, if $(f\oplus B)(x)=\bigvee_{b\in B}f(x-b)$, $(f\ominus B)(x)=\bigwedge_{b\in B}f(x+b)$, and $(f\circ B)(x)=(f\ominus B)\oplus B$ are the unit-scale Minkowski translation-invariant flat (i.e. unweighted) function operators, their multiscale counterparts are

$$\delta_{rB}(f)\triangleq f\oplus rB,\quad \varepsilon_{rB}(f)\triangleq f\ominus rB,\quad \alpha_{rB}(f)\triangleq f\circ rB \tag{7}$$

If B is *convex*, then [30]

$$rB=\underbrace{B\oplus B\oplus\cdots\oplus B}_{r\text{ times}},\quad r=0,1,2,... \tag{8}$$

This endows the above multiscale dilations and erosions with a semigroup property, which allows them to be generated recursively:

$$\delta_{(r+1)B}=\delta_B\delta_B^r,\quad \varepsilon_{(r+1)B}=\varepsilon_B\varepsilon_B^r,\quad r=0,1,2,... \tag{9}$$

and create the simplest case of a *morphological scale-space* [28, 11].

For *digital shapes and images*, the above translation-invariant morphological operators can be extended to multiple scales by using two alternative approaches. The first is an algebraic approach where if $B\subseteq\mathbb{Z}^d$ is a unit-scale discrete structuring graph, we define its scaled version rB for integer scales as in (8) and use (9) for producing multiscale morphological operators that agree with their continuous versions in (6) and (7) if B is convex. The second approach [1, 11] models the dilation and erosion scale-space functions $u(x,t)=f\oplus tB$ and $v(x,t)=f\ominus tB$ as generated by the nonlinear partial differential equations (PDEs)

$$\partial_t u=\|\nabla u\|_B,\quad \partial_t v=-\|\nabla v\|_B \tag{10}$$

where for a convex $B\subseteq\mathbb{R}^2$, $\|(x_1,x_2)\|_B=\sup_{(a_1,a_2)\in B}a_1x_1+a_2x_2$. These PDEs can be implemented using the numerical algorithms of [32], as explored in [35]. 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 detail by Kimia et al. [23]. Modern numerical algorithms for morphological PDEs can be found in [9, 10].

2.2 Background on Graph Morphology

We consider an undirected **graph** $G = (V,E)$ without 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 $E' \subseteq E$, the pair $G' = (V',E')$ is called a *subgraph* of G. A graph vertex mapping $\theta : V \to V'$ 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* $\mathrm{Sym}(G)$ of a graph. Symmetries play the role of 'generalized translations' on a graph.

Shapes $X \subseteq V$ and image functions $f : V \to \mathscr{T}$ defined on a graph G with values in a complete lattice $\mathscr{T}$ 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. $\mathscr{T} = \{0,1,...,m-1\}$, or continuous, e.g. $\mathscr{T} = \overline{\mathbb{R}}$. 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 (shape or function), i.e. $X \subseteq Y$ implies $\psi(X|G) \subseteq \psi(Y|G)$, and *G-increasing* if it increases in G, i.e., $G' \subseteq G$ implies $\psi(f|G') \leq \psi(f|G)$ for all graph functions $(f|G)$. A graph operator ψ is called *invariant* under graph symmetries $\tau \in \mathrm{Sym}(G)$ 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 $\mathscr{P}(V)$ can be defined via a **graph neighborhood function** $N : V \to \mathscr{P}(V)$ 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 (5) we also find its adjunct erosion:

$$\delta_N(X|G) \triangleq \bigcup_{v\in X} N(v), \quad \varepsilon_N(X|G) \triangleq \{v \in V : N(v) \subseteq X\} \tag{11}$$

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

$$\check{N}(v) \triangleq \{w \in V : v \in N(w)\} \tag{12}$$

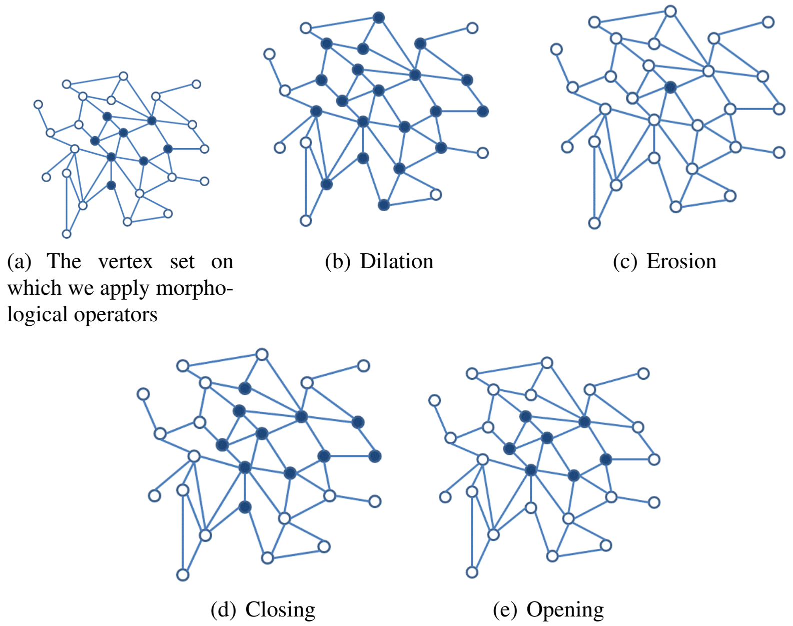
This is related to *operator duality* as follows. The dual (or negative) of a binary graph operator is defined by $\psi^*(X|G) = (\psi(X^*|G))^*$, where $X^* = V \setminus X$. Then, the dual graph dilation and erosion w.r.t. a neighborhood function coincide with the erosion and dilation, respectively, w.r.t. the reflected neighborhood function:

$$\delta_N^* = \varepsilon_{\check{N}}, \quad \varepsilon_N^* = \delta_{\check{N}} \tag{13}$$

If $N(v) = \check{N}(v)$ for each v, we have a *symmetric neighborhood* function. Such an example is Vincent's *unit-scale graph neighborhood* function [41]

$$N_1(v) \triangleq \{w \in V : (v,w) \in E\} \cup \{v\} \tag{14}$$

which, when centered at a vertex v, includes this vertex and all others that form an edge with it. If we use it in (11), 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 $\alpha_1 = \delta_1 \varepsilon_1$ and $\beta_1 = \varepsilon_1 \delta_1$ is a graph opening and closing, respectively. See Fig. 2 for an example. All four of these operators inherit the standard increasing property from their lattice definition and are invariant under graph symmetries. However, δ_1 is G-increasing, ε_1 is G-decreasing, and α_1 and β_1 are neither of these.


(a) The vertex set on which we apply morphological operators

(b) Dilation

(c) Erosion

(d) Closing

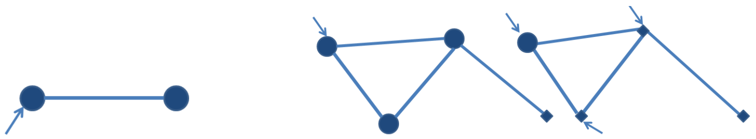
(e) Opening

Fig. 2 Binary graph operators using a unit-scale symmetric neighborhood function.

Heijmans et al. [20, 21] have generalized the above (symmetric neighborhood N_1) approach by introducing the concept of a **structuring graph *(s-graph)***. This is a graph $\mathscr{A} = (V_{\mathscr{A}}, E_{\mathscr{A}})$ of a relatively small size and has as additional structure two nonempty and possibly overlapping subsets: the *buds* $B_{\mathscr{A}} \subseteq V_{\mathscr{A}}$ and the *roots* $R_{\mathscr{A}} \subseteq V_{\mathscr{A}}$. 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 $\mathscr{A}$ corresponds to the following neighborhood function

$$N_{\mathscr{A}}(v|G) = \bigcup\{\theta(B_{\mathscr{A}}) : \theta \text{ embeds } \mathscr{A} \text{ into } G \text{ at } v\} \tag{15}$$

where we say that θ *embeds* $\mathscr{A}$ into G at v if θ is a group homomorphism of $\mathscr{A}$ into G and $v \in \theta(R_{\mathscr{A}})$. Such an embedding matches the s-graph $\mathscr{A}$ with the local structure of the graph G. The simple neighborhood N_1 of (14) corresponds to the s-graph of Fig. 3(a), with two vertices which both are buds and one of them is a root. Replacing (15) in (11) creates an adjunction of graph dilation and erosion $(\delta_{\mathscr{A}}, \varepsilon_{\mathscr{A}})$ 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., $N_{\mathscr{A}}(\tau(v)|G) = \tau N_{\mathscr{A}}(v|G)$, where $\tau X = \{\tau(v) : v \in X\}$.



(a) The s-graph that corresponds to the simple neighborhood. Specifically, using this s-graph as a strucuring 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.

Fig. 3 Examples of structuring graphs. Arrows indicate roots. Large circular nodes denote buds.

Finally, the reflection of the neighborhood of an s-graph equals the neighborhood of another s-graph $\check{\mathscr{A}}$, called the *reflection* of $\mathscr{A}$:

$$\check{N}_{\mathscr{A}}(v|G) = N_{\check{\mathscr{A}}}(v|G) \tag{16}$$

The reflected s-graph $\check{\mathscr{A}}$ has the same vertices and edges as the original s-graph $\mathscr{A}$ but their bud and root sets are interchanged: $B_{\check{\mathscr{A}}} = R_{\mathscr{A}}$ and $R_{\check{\mathscr{A}}} = B_{\mathscr{A}}$ (see Fig. 3). The dual operator of a dilation by an s-graph is the erosion by its reflected s-graph, and vice-versa, as prescribed by (13).

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 *level sets* $X_h(f|G) = \{v \in V : f(v) \geq h\}$, $h \in \mathscr{T}$:

$$(f|G)(v) = \sup\{h \in \mathscr{T} : v \in X_h(f|G)\} \tag{17}$$

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:

$$\psi(f|G)(v) = \sup\{h \in \mathscr{T} : v \in \psi(X_h(f)|G)\} \tag{18}$$

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

$$\delta_{\mathscr{A}}(f|G)(v) = \max_{w \in N_{\mathscr{A}}(v|G)} f(w) \tag{19}$$

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

2.3 Multiscale Morphology on Graphs

We need to discuss the notion of scale in graph morphology in order to obtain the graph counterparts of multiscale dilation and erosion defined in Section 2.1. Consider a graph $G = (V, E)$ and a nonempty subset $X \subseteq V$ of its vertices. Let $\mathscr{A}$ be an s-graph. One approach could be to define the dilation at scale $r = 1, 2, ...$ of a vertex subset X w.r.t. the s-graph $\mathscr{A}$ by $\delta_{r\mathscr{A}}(X|G)$ where $r\mathscr{A}$ denotes the r-fold graph dilation of the s-graph with itself. 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.

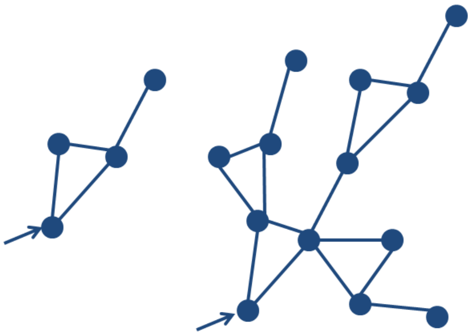


Fig. 4 Left: a structuring graph. Right: The scaled by $r = 2$ version of the s-graph. A scaling of a simple s-graph has increasingly complicated structure and therefore, for larger scales it is difficult or impossible to find an embedding to an arbitrary graph at each node. This fact necessitates an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

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

$$\delta^r_{\mathscr{A}}(X \mid G) = \delta_{\mathscr{A}}(\delta^{r-1}_{\mathscr{A}}(X \mid G) \mid G) \tag{20}$$

Essentially, we interchange the order with which we apply the dilation operators; in the classic framework in order to get the r-scale dilation we first find the r-scaling of the structuring graph and then perform the dilation with the set, whereas in our definition we dilate the set X with the structuring graph r times. Generalizing this notion of scale to multilevel dilation of a function $f : V \to \mathscr{T}$ we get the following definition. The dilation of f at integer scales r will be given at each $v \in V$ by

$$\phi_r(v) \triangleq \delta_{\mathscr{A}}(\phi_{r-1}(f \mid G) \mid G)(v), \quad \phi_1(v) = \delta_{\mathscr{A}}(f \mid G)(v) \tag{21}$$

This provides a recursive computation of the multiscale dilations of a function $f : V \to \mathscr{T}$ 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 $f : V \to \mathscr{T}$ by an s-graph $\mathscr{A}$ is described by the following difference equation at each vertex $v \in V$:*

$$\phi_{r+1}(v) - \phi_r(v) = \max_{w \in N_{\mathscr{A}}(v|G)} \{\phi_r(w) - \phi_r(v)\}. \tag{22}$$

Proof. By combining (21) with (19) we get

$$\phi_{r+1}(v) = \delta_{\mathscr{A}}(\phi_r(f \mid G))(v) = \max_{w \in N_{\mathscr{A}}(v|G)} \{\phi_r(w) - \phi_r(v)\} + \phi_r(v)$$

□

3 Geodesic Active Contours on Graphs

Kass et al. in [22] introduced the concept of energy minimizing snakes” driven by forces that pull it towards special features in the image like edges or lines. Specifically, the goal is to find in an image areas that are naturally distinguished from their background. The classical approach consists of a gradual deformation of an original curve towards the edges of those objects through the minimization, between successive time steps, of energy functionals which depend on the shape of the curve itself, its distance from the salient image features and finally terms that stabilize the snakes near local minima.

The main disadvantage of this initial approach is that 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 original snake will not be separated in two independent snakes. Heuristic solutions have been proposed in [31] but a topology-free approach has been given independently by Caselles et al. [12] and Malladi et al. [27]. These models are based on the theory of the curve evolution and geometric flows and the curve 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 boundary. Finally, the curve dynamics take the form of a geometric flow (PDE) and can be implemented conveniently using the level set methods proposed by Osher and Sethian [32] that can accommodate changes in the topology between successive curves.

In particular let $\mathbf{C}(q) : [0,1] \to \mathbb{R}^2$ be parameterized planar curve and let $I : [0,1]^2 \to \mathbb{R}_+$ be the image in which one needs to detect the objects' boundaries. Note that we denote the curve by $\mathbf{C}(\cdot)$ when we interpret it as a vector-valued function and by C when we interpret it as a set of points. The energy functional associated with C can be written as follows:

$$E(C) = \alpha \int_0^1 \|\mathbf{C}'(q)\|^2 dq - \lambda \int_0^1 g(\|\nabla I(\mathbf{C}(q))\|)\, dq \qquad a, \lambda \geq 0 \tag{23}$$

Let $g : [0, +\infty) \to \mathbb{R}_+$ be a strictly decreasing function such that $g(r) \to 0$ as $r \to \infty$. Caselles et al. in [13] show that the problem of finding the minimum energy curve as defined in (23) is equivalent to finding the minimum length curve in a Riemannian space induced from image I, whose length is given by

$$L_R = \int_0^1 g(\|\nabla I(\mathbf{C}(q))\|)\|\mathbf{C}'(q)\|\, dq = \int_0^{L(\mathbf{C})} g(\|\nabla I(\mathbf{C}(q))\|)\, ds, \tag{24}$$

where $L(C)$ is the Euclidean length of the curve C. Furthermore, it is shown that a curve which is governed from the dynamics

$$\frac{\partial \mathbf{C}(t)}{\partial t} = g(I) \cdot \kappa \cdot \mathbf{N} - (\nabla g \cdot \mathbf{N}) \cdot \mathbf{N} \tag{25}$$

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.

Assume now that the curve $C(t)$ is a level set of a function $u : \mathbb{R}^2 \times \mathbb{R}_+ \to \mathbb{R}$. Namely, $C(t)$ is the set of points $\mathbf{x}$ for which $u(\mathbf{x}, t)$ is equal to a constant (for example $u = 0$). It is shown that if the function $u(\mathbf{x}, t)$ satisfies

$$\frac{\partial u}{\partial t} = g(I)\|\nabla u\|(c + \kappa) + \nabla g(I) \cdot \nabla u, \tag{26}$$

then the corresponding level set satisfies (25).

Our goal is to approximate all terms in the right hand side of (26) on graphs and finally construct a difference equation which would approximate the active contour dynamics for edge detection. The next subsection is devoted to the analysis of the simplest case of curve evolution on graphs, that is the constant velocity motion introducing useful ideas from graph morphology operators. Observe that this case corresponds to approximating the first term of the RHS of (26). Subsequently, we will approximate the rest of the terms participating in the curve dynamics to end up with a geodesic active contour model on graphs.

3.1 Constant-Velocity Active Contours on Graphs

We derive the difference equation that describes the evolution of the contour of a set that expands with constant velocity on a graph. In the continuous case, a contour undergoing such an evolution corresponds to the boundary of the multiscale dilation of the set by a unit disk. If this set is given as a level set of a graylevel function $u : \mathbb{R}^2 \times \mathbb{R}_+ \to \mathbb{R}$, then the evolution of u is described by

$$\frac{\partial u}{\partial t} = \|\nabla u\|. \tag{27}$$

Consider a subset X of V. Let $\mathscr{A}$ be a structuring graph. Imitating the continuous case, the constant velocity expansion of X corresponds to its r-scale dilation, denoted by X_r. If X is given as the level set of an original graylevel function $u_0 : X \to \mathbb{R}$ and X_r is the level set of the r-scale graylevel function $u_r : X \to \mathbb{R}$, then the difference equation that governs u_r, by using Proposition 1, is

$$u_{r+1}(v) - u_r(v) = \max_{w \in N_{\mathscr{A}}(v|G)} \{u_r(w) - u_r(v)\}. \tag{28}$$

The above expression is a discrete equivalent of the gradient magnitude on graphs. Similar expressions are being used in the literature, [39, 14, 4]. Our work extends previous results to more general structuring elements and exploits the revealed insight to approximate other geometric properties of differential operators on graphs in the next sections. 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. Therefore, (29) is the graph counterpart of (27), which implies the approximation of $\|\nabla u\|$ at node v by $\max_{w \in N_{\mathscr{A}}(v|G)} \{u(w) - u(v)\}$.

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

1. Let u_0 be the signed distance function from X, defined by

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

   where X is the zero level set of u_0, and $d_E(w,v)$ corresponds to the Euclidean distance between the nodes w and v.
2. Evolve u_r according to the following, at scales $r = 1, 2, \ldots$

$$u_{r+1}(v) - u_r(v) = c \cdot \max_{w \in N_{\mathscr{A}}(v|G)} \{u_r(w) - u_r(v)\}. \tag{29}$$

3. The set $X_r = \{v \in X : u_r(v) \geq 0\}$ corresponds to the r-scale dilation of X.

Figure 5 illustrates the results for the constant velocity expansion of a circular contour. Throughout this chapter for all our simulations we are using the simple structuring graph of Fig. 3(a) that generates the simple neighborhood. Moreover we embed our shapes on 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 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 6000 to 10000 while ρ ranges from 0.015 to 0.04. Given the number of nodes, the parameter ρ affects the expected degree of a node and is proportional to the square root of its value.

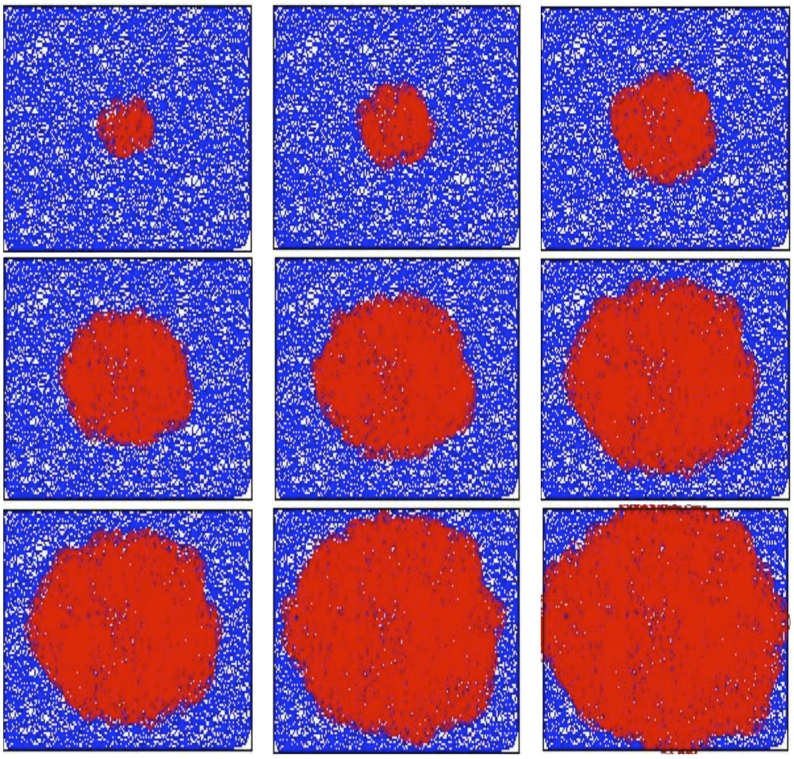


Fig. 5 Constant velocity evolution of a circular contour on a geometric random graph on the unit square. The structuring graph is an edge.

3.2 Direction of the Gradient on Graphs

Observing equation (26), in order to obtain a model for active contours on graphs, we need to define, except from the magnitude of the gradient vector which we have already done in the previous section, its direction and also a characterization for the curvature of the corresponding curve.

Beginning from the first, trusting our intuition from real analysis, 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. In other words, let a function u be defined on the set of nodes $v \in V$ of the graph $G = (V, E)$. Then

$$\frac{\nabla u}{\|\nabla u\|}(v) = \mathbf{e}_{v\hat{w}}, \quad \hat{w} = \mathrm{argmax}_{w \in N_{\mathscr{A}}(v|G)} \{u(w) - u(v)\}. \tag{30}$$

where $\mathbf{e}_{vw}$ is the unit vector in the direction of the edge (v,w). Although this approximation looks intuitive it does not work well in practice. In fact, consider the setting depicted in Fig. 6. Such a scenario is fairly usual in a graph structure due to its discrete nature. In other 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.

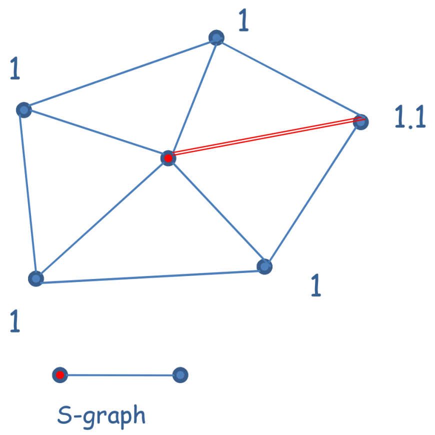


Fig. 6 Illustration of the disadvantages of choosing the maximum increase direction as the gradient direction. Observe that all directions yield approximately the same increase but only one edge is chosen. This happens exactly because of the discrete nature of the graph's structure. Alternatively, we propose a weighted average of edge directions where the weights are the function differences along each edge.

Taking the above into account we propose to approximate the gradient direction on graphs through a weighted average of the direction of all edges in a node's neighborhood, where weights will be the normalized corresponding differences, that is

$$\frac{\nabla u}{\|\nabla u\|}(v) \triangleq \frac{\sum_{w\in N_{\mathscr{A}}(v|G)}(u(w)-u(v))\mathbf{e}_{vw}}{\|\sum_{w\in N_{\mathscr{A}}(v|G)}(u(w)-u(v))\mathbf{e}_{vw}\|} \tag{31}$$

Finally, depending on the application, especially 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 control the influence of the edges with large increases. In those cases we may need to use the following expression as the direction of the gradient:

$$\frac{\nabla u}{\|\nabla u\|}(v) \triangleq \frac{\displaystyle\sum_{w\in N_{\mathscr{A}}(v|G)} \mathrm{sign}(u(w)-u(v))\frac{\mathbf{e}_{vw}}{\left(\max_{s\in N_{\mathscr{A}}(v|G)}\{\|u(s)-u(v)\|\}-\|u(w)-u(v)\|\right)^{p}+\varepsilon}}{\left\|\displaystyle\sum_{w\in N_{\mathscr{A}}(v|G)} \mathrm{sign}(u(w)-u(v))\frac{\mathbf{e}_{vw}}{\left(\max_{s\in N_{\mathscr{A}}(v|G)}\{\|u(s)-u(v)\|\}-\|u(w)-u(v)\|\right)^{p}+\varepsilon}\right\|} \tag{32}$$

Essentially, under (32) the edges along which the change in the value of the function is closer to the maximum contribute more to the gradient direction. By changing the parameter p we can adjust how strongly such edges affect the final outcome. Finally, ε is a small constant that guarantees that the denominators in (32) 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 $\varepsilon = 0.05$.

Having determined a meaning for the gradient of a function on a graph the only term remaining to be given a meaning on our more abstract graph structure is the curvature of the contour of each level set of the function u.

3.3 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

$$\kappa = \mathrm{div}\left(\frac{\nabla u}{\|\nabla u\|}\right). \tag{33}$$

On the other hand we have derived expressions for the term $\frac{\nabla u}{\|\nabla u\|}$ on a graph. Therefore, the remaining step is to propose an expression for the computation of the divergence of a function on a graph. Consider a vector function $\mathbf{F} : \mathbb{R}^2 \to \mathbb{R}^2$. The *divergence* of $\mathbf{F}$ at a point $\mathbf{x}$ is defined as

$$\mathrm{div}\mathbf{F}(\mathbf{x}) = \lim_{S\to\{\mathbf{x}\}} \frac{\oint_{\Gamma(S)} \mathbf{F}\cdot\mathbf{n}d\ell}{|S|}, \tag{34}$$

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.

To study the graph case consider Fig. 7. We can conclude that a good approximation for computing the divergence of $\mathbf{F}$ on a graph is the following

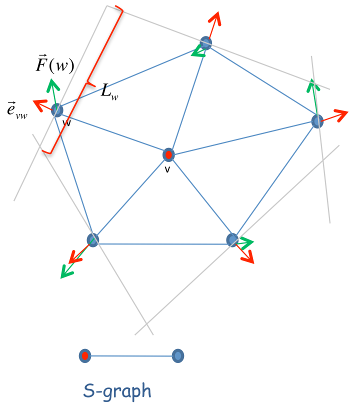
$$\mathrm{div}\mathbf{F}(v) = \frac{\sum_{w \in N_{\mathscr{A}}(v|G)} L_w \mathbf{F}(w) \cdot \mathbf{e}_{vw}}{S(v)} \tag{35}$$

where

- L_w corresponds to the length of the perpendicular to edge $\mathbf{e}_{vw}$,
- $S(v)$ corresponds to the area between the perpendicular to the edges lines,

as illustrated in Figure 7.

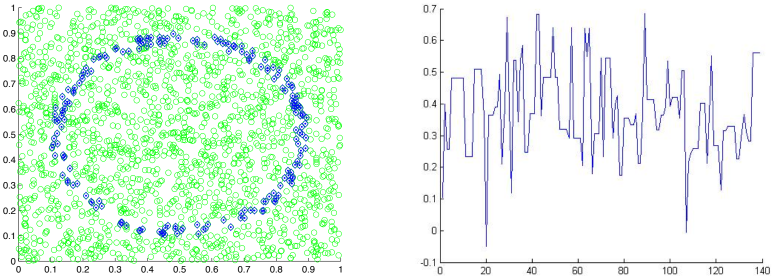
Fig. 7 Computing divergence on a graph: Let the green vectors denote the values of function $\mathbf{F}$, the red vectors be the unit vectors corresponding to each edge and let the gray lines be perpendicular to the corresponding edge.



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. 8. We would expect the curvature for all points on the circle to be a positive number, if we 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. 8.

3.4 Convolution on Graphs

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



(a) Circle on geometric graph.

(b) The curvature on the circle.

Fig. 8 The curvature on a circle calculated with the method proposed in Section 3.3. We omit the edges for illustration purposes.

following taken from [12, 27]:

$$g(\|\nabla I_\sigma\|) = \frac{1}{1 + \frac{\|\nabla I_\sigma\|^2}{\lambda^2}} \tag{36}$$

where,

$$I_\sigma = I * G_\sigma, \quad G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{37}$$

In order to compute the smoothed version I_σ of I we need to define the convolution operation on graphs. Let $G = (V,E)$ denote the underlying graph. Let $d_E(v,w)$ denote the Euclidean distance between vertices v and w. For each $v,w \in V$ define the function $G_\sigma^w(v)$ as follows:

$$G_\sigma^w(v) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{d_E(v,w)^2}{2\sigma^2}\right). \tag{38}$$

The smoothed version I_σ can be computed by mimicking the continuous convolution operation as follows:

$$I_\sigma(v) = \sum_{w \in V} I(v) G_\sigma^w(v). \tag{39}$$

3.5 Active Contours on Graphs - The Algorithm

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 $I : V \to \mathbb{R}$ 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 3.4.
2. Compute the magnitude of ∇I_σ as described in Section 3.1 and then compute the function $g(\|\nabla I_\sigma\|)$.
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 3.1 and 3.3. Iterate according to the following difference equation:

$$\phi_{r+1} - \phi_r = g(\|\nabla I_\sigma\|)\|\nabla\phi_{r-1}\|(c+\kappa) + g(\|\nabla I_\sigma\|)\cdot\nabla\phi_{r-1}, \quad c \geq 0 \quad (40)$$

Figure 9 illustrates the algorithm for the case of finding the boundaries of three disjoint objects (connected clusters of graph nodes).

4 Multiscale Skeletonization on Graphs

Since Blum's introduction of the skeleton or medial axis transform [5], it has received voluminous attention and has become one of the main tools for shape analysis and representation. The main process to find the skeleton is a distance wavefront propagation. In Euclidean spaces ($\mathbb{R}^d$, $d = 2,3$) this can be modeled either using a continuous distance formulation [5] or via continuous-space morphology [26, 36] or via PDEs that simulate these evolution operations [38, 3, 18, 40]. In the discrete 2D or 3D space $\mathbb{Z}^d$, the above approaches are replaced by discrete distance transforms and discrete morphology; for surveys and references see [34, 36]. The Chamfer distance transform is not always equivalent to the discrete morphology approach, unless the Chamfer ball is used as structuring element. Recent extensions of discrete distance transforms in 3D for skeletonization can be found in [2, 6]. One main disadvantage of the skeleton is its sensitivity on perturbations of the boundary. This can be partially addressed by using multiscale skeletons [29, 33], which provide a flexible framework of keeping only the skeleton parts that correspond to a smoothing of the original shape.

In this chapter we focus on the discrete skeleton transform obtained via multiscale morphology. For a discrete shape $X \subseteq \mathbb{Z}^2$ on regular grid, the morphological algorithm [29] computes the skeleton $S(X)$ of X, w.r.t. a disk-like unit-scale symmetric shape B,

$$S(X) = \bigcup_{n=0}^{N} S_n(X), \quad (41)$$

as a union of disjoint skeleton subsets $S_n(X)$, where

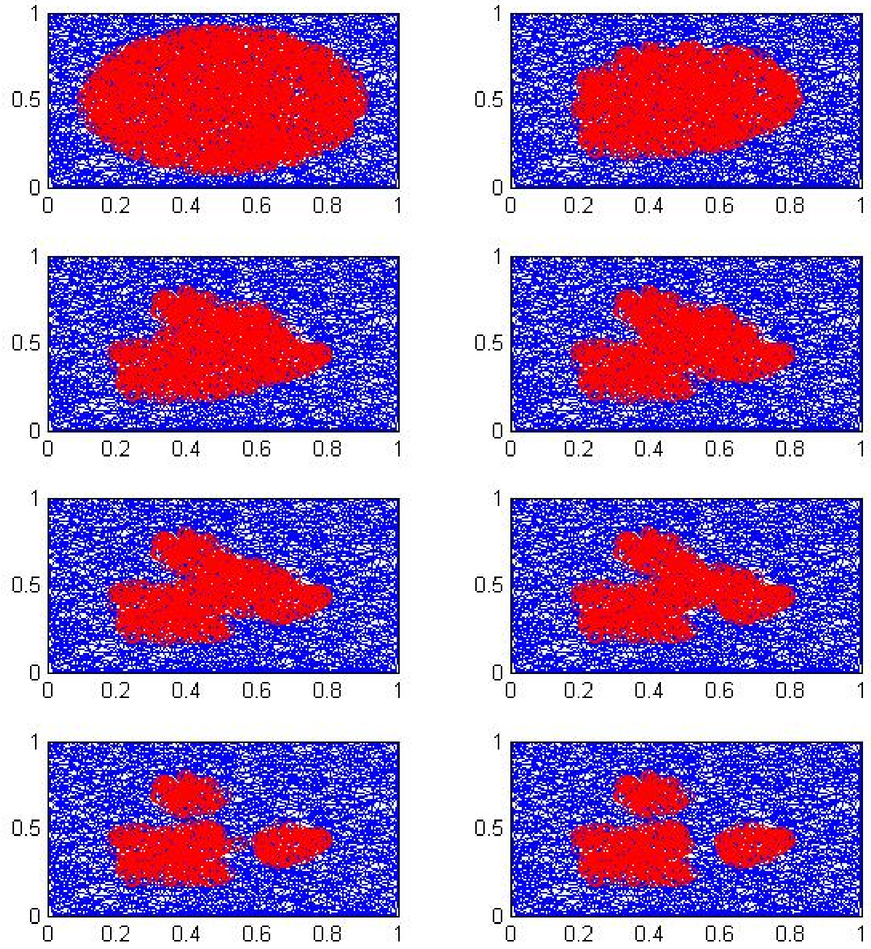


Fig. 9 Illustration of the active contour algorithms on graphs for finding three distinct objects on a graph. Note the change in the contour's topology. Time evolves from top left to bottom right.

$$S_n(X) = (X \oplus nB) \setminus [(X \ominus nB) \circ B], \tag{42}$$

indexed by the discrete scale $n = 0, 1, ..., N$, with $N = \max\{n : X \ominus nB \neq \emptyset\}$, where nB denotes the n-fold dilation of B with itself. Reconstruction of the original shape or its opening-smoothed versions requires the morphological skeleton transform $(S_0, S_1, ..., S_N)$, or equivalently the skeleton $S(X)$ and the quench function (the restriction of the distance transform onto the skeleton set):

$$X \circ kB = \bigcup_{n \geq k}^{N} S_n(X) \oplus nB, \quad k = 0, 1, 2, ... \tag{43}$$

Some generalizations of the above algorithm can be found in [25].

To extend multiscale skeletonization to shapes defined on arbitrary graphs $G = (V, E)$, we first provide a *lattice formulation* of the above discrete skeletonization algorithm adjusted for graph shapes. Let the shape be represented by a subset X of vertices of the graph and let $\mathscr{A}$ be a structuring graph. Then the skeleton of X can be obtained as follows:

$$S(X|G) \triangleq \bigcup_{n=0}^{N} S_n(X|G) \tag{44}$$

where

$$S_n(X|G) \triangleq \varepsilon^n_{\mathscr{A}}(X|G) \setminus \delta_{\mathscr{A}} \varepsilon^{n+1}_{\mathscr{A}}(X|G) \tag{45}$$

Taking the union of all or some of the skeleton subsets after dilating them in proportion to their scale yields respectively an exact or partial reconstruction of the graph shape:

$$\alpha_{k\mathscr{A}}(X|G) = \delta^k_{\mathscr{A}} \varepsilon^k_{\mathscr{A}}(X|G) = \bigcup_{n=k}^{N} \delta^n_{\mathscr{A}}[S_n(X|G)] \tag{46}$$

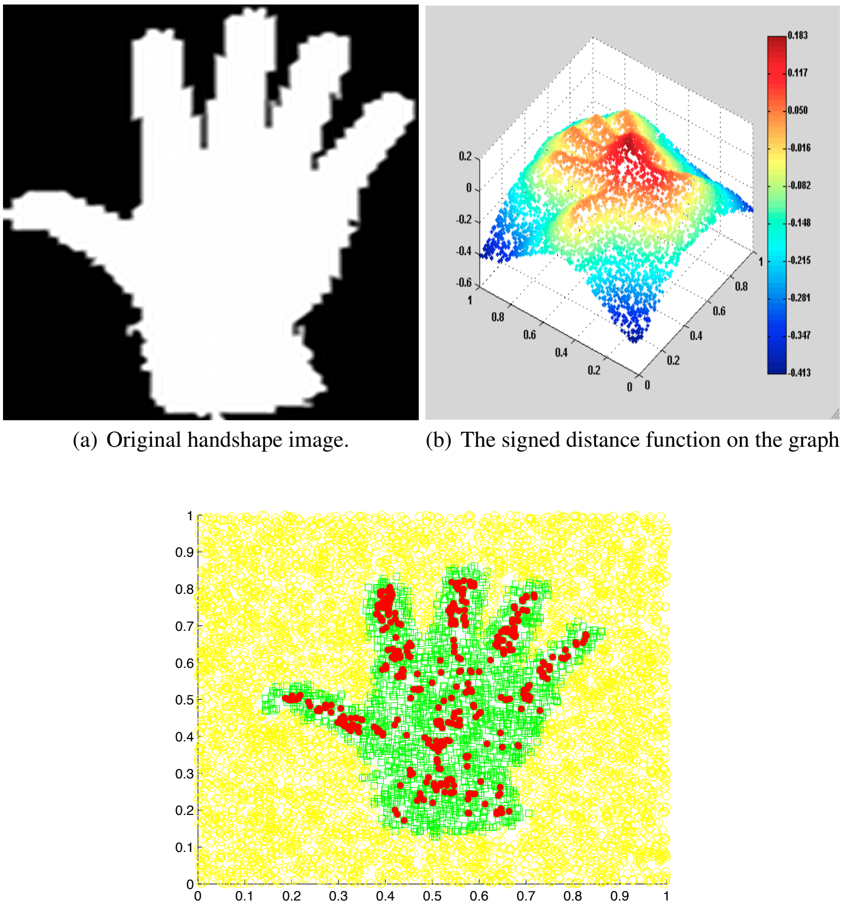
Namely, by not using the first k subsets, the above algorithm reconstructs the k-scale opening of the original graph shape.

Next we explore the application of the difference equation based techniques that we developed in order to calculate the skeleton of a shape defined on graphs. Specifically, we propose a calculation of the multiscale graph dilations and erosions involved using the active contour and level set approximations that we introduced in Section 3. The main idea can be summarized in the following:

Algorithm - Skeleton Calculation

```
Initialization:
$u(\cdot) = d_{sgn}(\cdot \mid X)$ (X is the shape whose skeleton we are computing)
$S \rightarrow \emptyset$
execution:
while $\max_{v \in G} u(v) > 0$ do
    $u'(v) = \min_{w \in N_{\mathscr{A}}(v)} u(w)$                (erosion-$\varepsilon^n_{\mathscr{A}}(X)$)
    $r(v) = \min_{w \in N_{\mathscr{A}}(v)} u'(w)$                (erosion- $\varepsilon^{n+1}_{\mathscr{A}}(X)$)
    $o(v) = \max_{w \in N_{\mathscr{A}}(v)} r(w)$                (dilation- $\delta_{\mathscr{A}} \varepsilon^{n+1}_{\mathscr{A}}(X)$)
    $S \leftarrow S \cup \{v : u'(v) \geq 0\} \setminus \{v : o(v) \geq 0\}$     (set difference)
    $u(v) = u'(v)$
end while
```

The results of our graph skeleton algorithm, simulated on a handshape object represented by a geometric random graph, are presented in Fig. 10.



(a) Original handshape image.

(b) The signed distance function on the graph

(c) The calculated skeleton.

Fig. 10 The skeleton of a handshape image calculated using our algorithm. The green squares correspond to the embedding of the handshape image on the geometric random graph. The red dots correspond to the skeleton transform. The yellow circles are irrelevant nodes of the underlying graph.

5 Conclusions

In this chapter we have proposed an approximation to level set implementation of morphological operators, skeleton transforms 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.

In our simulations we have mainly assumed geometric random graphs and simple structuring graphs. The choice of the simple s-graph is reasonable for any underlying graph structure with no prior information on the graph's characteristics. It is of great interest to other applications to correlate information on the underlying graph with the choice of the structuring graph.

Moreover, we are proposing approximations to concepts from calculus and there is space for better heuristics and modifications to this end. Keeping in mind the original energy minimization approach for geodesic active contours instead of the analytic solution, by properly defining and calculating an energy that corresponds to each contour and applying a step by step minimization procedure we obtain another approach to geodesic active contours. On the other hand, since the latter involves a minimization problem at each time step it is computationally less efficient but it may yield more accurate results.

Finally, regarding the application of the above ideas that we have introduced to multiscale shape skeletonization on arbitrary graphs, one research direction of interest is how to analyze or guarantee the connectedness of the resulting skeleton.

References

1. L. Alvarez, F. Guichard, P.L. Lions, and J. M. Morel. Axioms and Fundamental Equations of Image Processing. *Archiv. Rat. Mech.*, 123(2):199–257, June 1993.
2. C. Arcelli, G. Sanniti di Baja, and L. Serino. Distance-Driven Skeletonization in Voxel Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(4):709–720, Apr. 2011.
3. C. Aslan, A. Erdem, E. Erdem, and S. Tari. Disconnected Skeleton: Shape at Its Absolute Scale. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(12):2188–2203, Dec. 2008.
4. A. Bensoussan and J. Menaldi. Difference Equations on Weighted Graphs. *J. Convex Analysis*, 12(1):13–44, 2005.
5. H. Blum. A Transformation for Extracting New Descriptors of Shape. In *Proc. Symposium on Models for the Perception of Speech and Visual Forms, Boston, Nov. 1964*. MIT Press, 1967.
6. G. Borgefors, I. Nyström, and G. Sanniti di Baja. Discrete Skeletons from Distance Transforms in 2D and 3D. In K. Siddiqi and S. Pizer, editors, *Medial Representations Mathematics, Algorithms and Applications*. Springer, 2008.
7. Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, Sep. 2004.
8. Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.
9. M. Breuß and J. Weickert. A Shock-Capturing Algorithm for the Differential Equations of Dilation and Erosion. *J. Math. Imaging and Vision*, 25:187–201, 2006.

10. M. Breuß and J. Weickert. Highly Accurate Schemes for PDE-Based Morphology with General Convex Structuring Elements. *Int. J. Comput. Vis.*, 92(2):132–145, 2011.
11. R. W. Brockett and P. Maragos. Evolution Equations for Continuous-Scale Morphological Filtering. *IEEE Trans. Signal Processing*, 42(12):3377–3386, Dec. 1994.
12. V. Caselles, F. Catte, T. Coll, and F. Dibos. A Geometric Model for Active Contours in Image Processing. *Numerische Mathematik*, 66(1):1–31, 1993.
13. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. *Int. J. Comput. Vis.*, 22(1):61–79, 1997.
14. S.-Y. Chung and C. A. Berenstein. Omega-Harmonic Functions and Inverse Conductivity Problems on Networks. *SIAM J. Applied Mathematics*, 65:1200–1226, 2005.
15. T.F. Cootes, G. J. Edwards, and C.J. Taylor. Active Appearance Models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.
16. C. Couprie, L. Grady, L. Najman, and H. Talbot. Power Watershed: A Unifying Graph-based Optimization Framework. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(7):1384–1399, July 2011.
17. J. Cousty, L. Najman, and J. Serra. Some morphological operators in graph spaces. In *Proc. Int. Symp. Mathematical Morphology*, LNCS 5720. Springer, 2009.
18. M. S. Hassouna and A. A. Farag. Variational Curve Skeletons Using Gradient Vector Flow. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(12):2257–2274, Dec. 2009.
19. H.J.A.M. Heijmans. *Morphological Image Operators*. Acad. Press, Boston, 1994.
20. H.J.A.M. Heijmans, P. Nacken, A. Toet, and L. Vincent. Graph Morphology. *J. Vis. Comun. Image Represent.*, 3(1):24–38, 1992.
21. H.J.A.M. Heijmans and L. Vincent. Graph Morphology in Image Analysis. In E.R. Dougherty, editor, *Mathematical Morphology in Image Processing*. Marcel Dekker, NY, 1993.
22. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *Int. J. Comput. Vis.*, 1(4):321–331, 1988.
23. 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.*, 163:438–458, 1992.
24. N. Komodakis, N. Paragios, and G. Tziritas. MRF Energy Minimization and Beyond via Dual Decomposition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33:531–552, Mar. 2011.
25. R. Kresch and D. Malah. Skeleton-based Morphological Coding of Binary Images. *IEEE Trans. Image Processing*, 7(10):1387–1399, Oct. 1998.
26. C. Lantuejoul. Skeletonization in Quantitative Metallography. In R.M. Haralick and J.C. Simon, editors, *Issues of Digital Image Processing*. Sijthoff and Noordhoff, Groningen, The Netherlands, 1980.
27. R. Malladi, J. A. Sethian, and B. C. Vemuri. A Fast Level Set based Algorithm for Topology-Independent Shape Modeling. *J. Math. Imaging and Vision*, 6:269–289, 1996.
28. P. Maragos. Pattern Spectrum and Multiscale Shape Representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11:701–716, July 1989.
29. P. Maragos and R. W. Schafer. Morphological Skeleton Representation and Coding of Binary Images. *IEEE Trans. Acoustics, Speech and Signal Processing*, 34:1228–1244, October 1986.
30. G. Matheron. *Random Sets and Integral Geometry*. Wiley, New York, 1975.
31. T. McInerney and D. Terzopoulos. Topologically Adaptable Snakes. In *Proc. Int'l Conf. on Computer Vision*, pages 840–845, 1995.
32. S. Osher and J. A. Sethian. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *J. Comp. Phys.*, 79:12–49, 1988.
33. S. M. Pizer, K. Siddiqi, G. Székely, J. N. Damon, and S. W. Zucker. Multiscale Medial Loci and Their Properties. *Int. J. Comput. Vis.*, 55(2/3):155–179, 2003.
34. A. Rosenfeld and A. C. Kak. *Digital Picture Processing: Vol I and II.* Acad. Press, 1982.
35. G. Sapiro, R. Kimmel, D. Shaked, B. Kimia, and A. Bruckstein. Implementing Continuous-scale Morphology via Curve Evolution. *Pattern Recognition*, 26(9):1363–1372, 1993.
36. J. Serra. *Image Analysis and Mathematical Morphology*. Acad. Press, 1982.
37. J. Serra, editor. *Image Analysis and Mathematical Morphology*, volume 2: Theoretical Advances. Acad. Press, 1988.

38. K. Siddiqi, S. Buix, A. Tannenbaum, and W. Zucker. Hamilton-Jacobi Skeletons. *Int. J. Comput. Vis.*, 48(3):215–231, 2002.
39. V.-T. Ta, A. Elmoataz, and O. Lézoray. Nonlocal PDEs-Based Morphology on Weighted Graphs for Image and Data Processing. *IEEE Trans. Image Processing*, 20(6):1504–1516, June 2011.
40. S. Tari. Hierarchical Shape Decomposition via Level Sets. In *Proc. Int. Symp. Mathematical Morphology*, LNCS 5720. Springer, 2009.
41. L. Vincent. Graphs and Mathematical Morphology. 16(4):365–388, 1989.
42. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms. *IEEE Trans. Info. Theory*, 51(7):2282–2312, July 2005.