Accurate Spatial Neighborhood Relationships for Arbitrarily-Shaped Objects Using Hamilton-Jacobi GVD*

Sumit K. Nath, Kannappan Palaniappan, and Filiz Bunyak

MCVL, Department of Computer Science, University of Missouri-Columbia, MO, USA
{naths, palaniappank, bunyak}@missouri.edu

Abstract. Many image segmentation approaches rely upon or are enhanced by using spatial relationship information between image regions and their object correspondences. Spatial relationships are usually captured in terms of relative neighborhood graphs such as the Delaunay graph. Neighborhood graphs capture information about which objects are close to each other in the plane or in space but may not capture complete spatial relationships such as containment or holes. Additionally, the typical approach used to compute the Delaunay graph (or its dual, the Voronoi polytopes) is based on using only the point-based (i.e., centroid) representation of each object. This can lead to incorrect spatial neighborhood graphs for sized objects with complex topology, eventually resulting in poor segmentation. This paper proposes a new algorithm for efficiently, and accurately extracting accurate neighborhood graphs in linear time by computing the Hamilton-Jacobi generalized Voronoi diagram (GVD) using the exact Euclidean-distance transform with Laplacian-of-Gaussian, and morphological operators. The algorithm is validated using synthetic, and real biological imagery of epithelial cells.

1 Introduction

Spatial neighborhood relationships among objects is an important characteristic in many image analysis, computer vision and robotics applications. One common approach is to compute Delaunay graphs from an ordinary Voronoi diagram (OVD), using information from centroids of objects [1]. In the context of biological image analysis, the OVD has been used for accurate segmentation and analysis of confluent migrating cells [2, 3], tissue architecture characterization [4], or endothelial cell classification [5]. Our application is primarily focused on accurate segmentation and tracking of cells in biomedical video sequences that undergo complex shape changes like mitosis and apoptosis.

An OVD using points is insensitive to object properties like size, shape, orientation or containment. Thus, neighborhood graphs derived from point-based centroid representations of arbitrarily-shaped objects often lead to incorrect neighborhood relationships as shown in Figs. 1(b) and (e). Applications that depend on accurate spatial neighborhood relationships would consequently fail or lead to unpredictable behavior. For example, incorrect neighborhood relationships may lead to false merges of neighboring cells in the segmentation algorithm described in [2]. In other applications, such as robot path

* This work was supported by a U.S National Institute of Health NIBIB award R33 EB00573.

B.K. Ersbøll and K.S. Pedersen (Eds.): SCIA 2007, LNCS 4522, pp. 421–431, 2007.

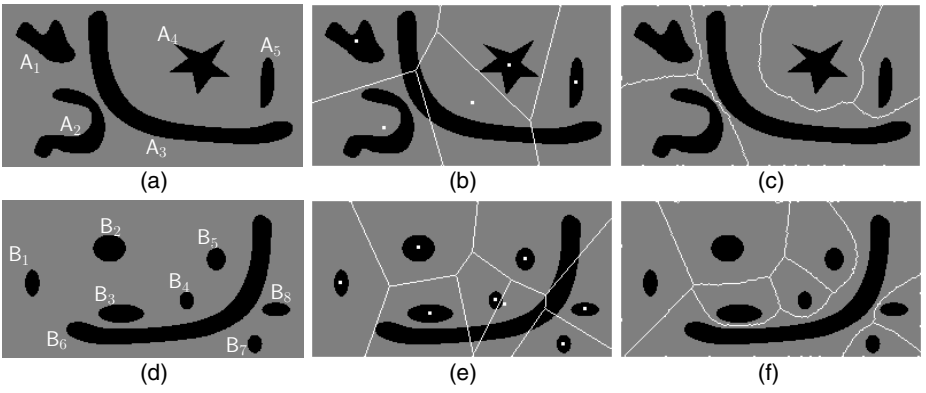

Fig. 1. [**a** and **d**]: Synthetic images showing arbitrarily-shaped objects. [**b** and **e**]: It is evident that an ordinary Voronoi diagram (OVD), computed from centroid points of objects (shown as white squares) leads to incorrect neighborhood relationships. [**c** and **f**]: However, using a generalized Voronoi diagram (GVD) leads to correct boundaries and neighborhood relationships. Corresponding neighborhood adjacency graphs are shown in Tables 1 and 2.

planning, inaccurate neighborhoods obtained from OVD's may impede the movement of the robot or lead to weak navigation performance [6]. An alternative to the OVD is to compute the generalized Voronoi diagram (GVD) that takes into account the size, shape, orientation, and placement of objects when computing neighborhood relationships. As seen from Figs. 1(c) and 1(f) the GVD accurately identifies the neighborhoods of complex-shaped objects (e.g., the thin long non-convex worm-like object).

The GVD in any dimension can be precisely defined using point to object distance measures [1, p 280]. Let $\mathbf{A} = \{A_1, A_2, \ldots, A_N\}$ be a set of arbitrarily shaped objects in a $d-$dimensional space $\mathbb{R}^d$. Now, for any point $\mathbf{p} \in \mathbb{R}^d$, let $\mathsf{D}(\mathbf{p}, A_i)$ denote a distance measure representing how far the point $\mathbf{p}$ is from the object A_i which is typically the minimum distance from $\mathbf{p}$ to any point in object A_i. The dominance region (also known as influence-zone) of A_i, is then defined as

$$\mathrm{Dom}(A_i, A_j) = \Big\{ \mathbf{p} \,|\mathsf{D}(\mathbf{p}, A_i) \leq \mathsf{D}(\mathbf{p}, A_j),\, \forall j, j \neq i \Big\} \tag{1}$$

A generalized Voronoi boundary, between A_i and Aj, can then be defined as the loci of equidistant points between both objects, $\mathcal{L}(A_i, A_j)$, where

$$\mathcal{L}(A_i, A_j) = \Big\{ \mathbf{p} \,|\mathsf{D}(\mathbf{p}, A_i) = \mathsf{D}(\mathbf{p}, A_j) \Big\}, \tag{2}$$

and the corresponding influence zone for A_i, $V(A_i)$, is the set intersection

$$V(A_i) = \bigcap_{i \neq j} \mathrm{Dom}(A_i, A_j) \tag{3}$$

Hence, the generalized Voronoi diagram of $\mathbf{A}$, $\mathrm{GVD}(\mathbf{A})$, is given by the union of of such generalized Voronoi regions, as

$$\begin{aligned}\mathrm{GVD}(\mathbf{A}) &= \bigcup_i V(A_i) \\ &= \bigcup_i \bigcap_{i \neq j} \left\{ \mathbf{p} \,|\mathsf{D}(\mathbf{p}, A_i) \leq \mathsf{D}(\mathbf{p}, A_j),\, \forall j, j \neq i \right\} \end{aligned} \tag{4}$$

When $\mathbf{A}$ is a collection of points rather than sized objects, $\mathrm{GVD}(\mathbf{A})$ reduces to an ordinary Voronoi diagram, $\mathrm{OVD}(\mathbf{A})$. Note that OVD boundaries are always straight lines or hyperplanes, whereas GVD boundaries can be complex curves or surfaces. Fig. 1 shows examples of the OVD and GVD for objects in a plane (i.e., $d = 2$). For those interested in properties of the OVD for point objects, we direct them to the book by Okabe *et al.* [1] and the survey paper by Aurenhammer [7].

The GVD representation of a set of objects has a number of useful properties: (i) it is a thin set that partitions a space into connected regions (ii) it is homotopic to the number of objects, (iii) it is invariant under transformations applied to all objects, and (iv) each region of the GVD is guaranteed to contain the entire object.

Sugihara presents an algorithm to construct an approximate GVD by reducing an object to a collection of points [8]. A different class of algorithms to construct GVD's is based on morphological operators and label propagation. This consists of labeling connected components (objects) in an image, and simultaneously growing them using dilation operators. The loci of points at which these regions stop growing determine the influence zone of each object. In the literature, this algorithm is referred to as *skeletons by influence zone* (SKIZ) and is described in detail by Vincent [9, 5]. Lu and Tan have presented a variation of SKIZ by approximating connected components as polygons and expanding the regions using Freeman codes for document image analysis [10]. Hoff *et al.* have reported a fast algorithm for GVD construction using graphics hardware [11].

Recently, Siddiqi *et al.* proposed a new class of algorithms to compute object skeletons using the average outward flux of the gradient of a distance transform [12]. Homotopy preserving properties of this algorithm makes it a strong alternative to other algorithms that use the Euclidean distance transform (EDT) to compute object skeletons. A Hamilton-Jacobi formulation for *shock tracking*, combined with homotopy preserving thinning leads to a robust and low-complexity implementation. As an original contribution, we propose using the Hamilton-Jacobi formulation to compute GVD's. The focus of this paper is on efficiently extracting exact neighborhood relationships of arbitrarily shaped objects (e.g., biological cells) using the GVD as the basic underlying framework, based on a fast EDT. It should be noted that even though our algorithm aims at solving a problem in biological image analysis, it can be applied to other applications in computer vision such as robot navigation, remote sensing of urban areas or content-based image retrieval.

The paper is organized as follows. In Sec. 2, we summarize our proposed algorithm and explain its key features. Comparative results of using OVD versus GVD for computing cell neighborhood relationships are shown in Sec. 3, and conclusions in Sec. 4.

Algorithm 1. `Compute a 2D Neighborhood Adj. Graph`

Input : $\mathbf{P}$, a 2D mask with N labeled objects,
T_{LD}, threshold to detect ridges,
T_{HS}, threshold for max. hole size, and
σ, to control smoothing.

Output : $\mathcal{N}(\mathbf{P})$, the adjacency graph of $\mathbf{P}$

1: Remove labeled 8-connected pixels in $\mathbf{P}$ that are adjacent to one or more different labels.
2: Convert the processed mask into a binary image $\mathcal{B}$.
3: Compute the Euclidean distance transform (EDT), $\mathcal{D}$, of $\mathcal{B}$ using the FH-EDT algorithm [13].
4: Compute $\mathsf{E} = \nabla^2 G_\sigma \circledast \mathcal{D}$, the Laplacian of the smoothed EDT.
5: Obtain a binary image, E_{thr}, from E using a threshold value T_{LD}.
6: Fill holes using T_{HS}, the hole-size threshold.
7: Apply a suitable thinning algorithm (e.g., [14, 15]) on E_{thr} to obtain an image with 1-pixel thick GVD boundaries, E_{thr}^{thn}.
8: Apply any homotopy-preserving algorithm [16] to prune *branches* from the generalized Voronoi diagram, E_{thr}^{thn}.
9: Assign $\mathcal{Q} \leftarrow (\mathsf{E}_{thr}^{thn})^c$; the complementary image of E_{thr}^{thn}
10: Using 4-connectivity, label the connected components of $\mathcal{Q}$.
11: Update the neighborhood relationship map $\mathcal{N}(\mathbf{P})$ by checking a 3×3 neighborhood of each background pixel (i.e., boundary pixels of connected components) in $\mathcal{Q}$.

2 Neighborhood Adjacency Graphs Using GVD

The proposed algorithm to compute a neighborhood adjacency graph $\mathcal{N}(\mathbf{P})$ for an image $\mathbf{P}$ containing $N-$arbitrarily shaped objects, using GVD in $\mathbb{R}^2$ is shown in Algorithm 1., and described in detail in the following paragraphs.

In order to compute reliable GVD boundaries touching objects need to be separated by at least a one-pixel gap. In Step 1, labeled pixels are (temporarily) removed from the image if they are adjacent to one or more different labeled pixels, without any gap. In Step 2, we convert the modified multi-labeled mask into a binary image with non-zero pixels representing N distinct connected components.

Siddiqi *et al.* have reported using a Borgefors distance transform (BDT) in their skeletonization algorithm [12]. However, the BDT is an approximation of the Euclidean distance transform (EDT). Hence, in Step 3, we compute the exact EDT using a "separable algorithm" proposed by Felsenzwalb and Huttenlocher (FH-EDT) that is fast (linear time), and efficient to implement [13].

Let $\mathcal{G}_1 = \{0, 1, \ldots, n-1\}$ be a 1D grid, and $f : \mathcal{G}_1 \rightarrow \mathbb{R}$ an arbitrary function on the grid. The one-dimensional FH-EDT of f is defined as

$$\mathcal{D}_f(p) = \min_{q \in \mathcal{G}_1} \Big((p-q)^2 + f(q) \Big) \tag{5}$$

with the added constraint that for each point $q \in \mathcal{G}_1$, the distance transform of f is bounded by a parabola rooted at $(q, f(q))$. The distance transform at point p is the height of the lower envelope of all such parabolas [13, Fig. 1]. The FH-EDT algorithm

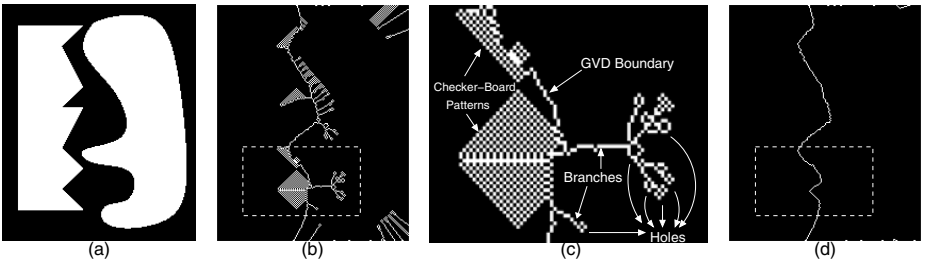

Fig. 2. Examples of isolated holes and checker-board pattern holes that are formed when a large number of single pixel width ridges appear very close to each other. Non-pruned branches may affect the performance of such applications as robot navigation. A relatively higher value of T_{LD} can reduce such occurrences, with the possibility of breaking actual GVD boundaries. This figure is related to problems that are solved in Steps 6 - 8 in Algorithm 1.. The actual GVD boundary is shown in (d).

computes the distance transform in $O(n)$ time. The efficiency of this algorithm is evident by considering a two-dimensional grid $\mathcal{G}_2 = \{0, 1, \ldots, n-1\} \times \{0, 1, \ldots, m-1\}$, and $f : \mathcal{G}_2 \rightarrow \mathbb{R}$ an arbitrary function on the grid. The two-dimensional distance transform of f is given by

$$\begin{aligned} \mathcal{D}_f(x, y) &= \min_{x', y'} \Big((x - x')^2 + (y - y')^2 + f(x', y'), \Big) \\ &= \min_{x'} \Big((x - x')^2 + \min_{y'} \Big((y - y')^2 + f(x', y') \Big) \Big), \\ &= \min_{x'} \Big((x - x')^2 + \mathcal{D}_{f|_{x'}}(y) \Big), \end{aligned} \tag{6}$$

where $\mathcal{D}_{f|_{x'}}(y)$ is the 1D distance transform of f restricted to the column indexed by x'. Hence, the 2D distance transform can be computed separably in linear time.

In order to detect points of singularities (or *shock points*), Siddiqi *et al.* propose to compute the average outward flux at every point in a vector field $\dot{\mathbf{q}}$ (derived from the distance transform) using a Hamilton-Jacobi formulation [12]. Using the divergence theorem, a relationship between the divergence of the vector field $\mathrm{div}(\dot{\mathbf{q}})$, and the average outward flux is given by [12]

$$\mathrm{div}(\dot{\mathbf{q}}) \equiv \lim_{\Delta a \rightarrow 0} \frac{\int_{\delta R} < \dot{\mathbf{q}}, \mathcal{N}_s > \; ds}{\Delta a}, \tag{7}$$

where δR is the bounding contour of the region R, $\mathcal{N}_s$ is the outward normal at each point of the contour, and ds is the element of integration. The divergence $\mathrm{div}(\dot{\mathbf{q}})$ can be equivalently written as the sum of partial derivatives with respect to each of the vector field's component directions. However, the vector field (i.e., distance field) is differentiable at all points except at *singular or shock points*. This is the justification provided by Siddiqi *et al.* for using Eq. 7, and a limit approximation, to locate singularities in $\dot{\mathbf{q}}$. As an alternative, in Step 4, we propose using a 2D Laplacian-of-Gaussian

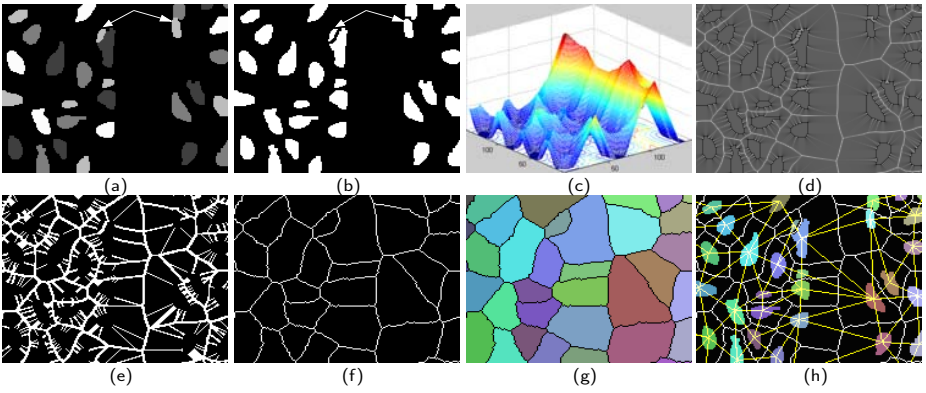

Fig. 3. A flow diagram that describes Algorithm 1.. (a) A section of the original mask with four unique foreground colors obtained from Nath *et al.* algorithm [2]. Cells that are touching each other are marked with arrows. (b) A binary image $\mathcal{B}$ is obtained as per rules outlined in as per Step 1 of Algorithm 1.. (c) 3D view of the distance transform ($\mathcal{D}$), that shows the difficulty in isolating ridges (i.e., Voronoi boundaries). (d) $\nabla^2 \mathcal{D}_\sigma$. (e) A thresholded version of Fig. 3(d) after removal of small holes. (f) A pruning step removes *branches* from the generalized Voronoi diagram. (g) Connected-component labeling, followed by generation of $\mathcal{N}(\mathbf{P})$, is implemented on a complement of the image, obtained in Fig 3(f), as per Steps 10-11 of Algorithm 1.. (h) The final generalized Voronoi diagram and neighbors of cells are shown in white and yellow, while cells are shown in colors used previously for labeling Voronoi cells in Fig. 3(g).

$\nabla^2 G_\sigma \circledast \mathcal{D}$ operator on the distance transform, $\mathcal{D}$, in order to detect regions of local maxima (or minima, depending on how the Laplacian operator is applied), i.e., ridge points. The Gaussian operator G_σ smooths the distance transform prior to applying the Laplacian operator insuring differentiability at shock points. Smoothing, however, does not guarantee homotopy preservation of GVD boundary points. Hence, to satisfy both constraints, the regularization parameter σ is set to a small value.

In Step 5, we threshold $\mathsf{E} = \nabla^2 G_\sigma \circledast \mathcal{D}$) to obtain the binary image, E_{thr},

$$\mathsf{E}_{thr} = \begin{cases} 1 & \mathsf{E} > T_{LD}, \\ 0 & \text{otherwise}, \end{cases}$$

before computing the GVD, A suitable choice of the threshold value, T_{LD}, is critical in homotopy preservation of GVD boundaries. A low threshold value results in larger number of spurious features (such as branches and associated holes), while a larger threshold significantly reduces these features at the cost of breaking real object boundaries. We set $T_{LD} = 0$ by default.

After binarization of E, the background *should* normally be segmented into N connected generalized Voronoi regions, corresponding to N input objects. However, when computing the Laplacian of the EDT, regions of local maxima, i.e., ridges, may appear very close to each other and interact to produce "*holes*" that are small connected background components (shown in Fig. 2(b) and (c)). In our algorithm, each influence zone (i.e., $V(A_i)$) corresponds to a unique object (A_i) in the image. Hence, in Step 6, such

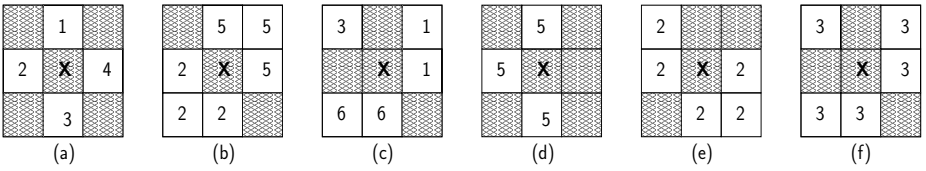

Fig. 4. Examples of neighborhood relationships between connected components when centering a 3×3 neighborhood on a boundary pixel (marked with a X). Shaded regions indicate one pixel thick boundaries of connected components in $\mathcal{Q}$. Valid generalized Voronoi boundary pixels separate different influence zones, resulting in *at least* two different sets of foreground labels in the neighborhood (Figs. 4(a) - 4(c)). On the other hand, spurs/branches are contained within a single influence zone, thus resulting in a single set of foreground labels in the neighborhood. As a result, no changes need to be made in the adjacency graph $\mathcal{N}(\mathbf{P})$ (Figs. 4(d) - 4(f)).

holes are removed using a threshold parameter T_{HS}, prior to computing the GVD. Non-removal of such holes prevents further removal of ridges that are attached to such holes, termed as *branches*. Hole removal is effected by size-constrained connected component analysis. The binarized image, obtained in Step 5 of Algorithm 1. is inverted followed by a connected component analysis. All connected components below a certain size are classified as part of the background which results in "hole-filling".

In Step 8, a thinning algorithm (c.f. [15]) is applied to the hole-filled, binarized image in order to reduce ridge boundaries to single pixel thickness. This step is necessary in order to simplify the search for neighborhood adjacency relationships along boundaries. A key component of any thinning algorithm is the preservation of end points. Thus, after thinning, spurious ridges, without holes, remain attached to actual GVD boundaries. We term such ridges as *spurs* (see Fig. 5(f) for example). Hence, in Step 9, we remove such spurs by applying a pruning algorithm having the same features as standard thinning algorithms (e.g., [15]) but enforcing the constraint of non-preservation of end points. Let this thinned (and optionally pruned) image be represented as E_{thr}^{thn}.

After obtaining one-pixel thick GVD boundaries, we invert E_{thr}^{thn} in Step 9 as $\mathcal{Q} = (\mathsf{E}_{thr}^{thn})^c$. This is followed, in Step 10, by a connected component analysis on $\mathcal{P}$ and assigning unique labels to each GVD influence zone, i.e., $\mathbf{Q} = \bigcup_i \mathcal{Q}(V_i)$, where $\mathcal{Q}(V_i)$ is the i^{th} connected component formed from the corresponding generalized Voronoi influence zone. Finally, in Step 11, a 3×3 window positioned at each boundary pixel (i.e., pixels not part of any connected component) is analyzed, from which a neighborhood relationship map $\mathcal{N}(\mathbf{P})$ is constructed (see Fig. 4 for some examples). To complement the discussion in previous paragraphs, key steps of our algorithm are shown in Fig. 3.

3 Results and Discussion

The Hamilton-Jacobi GVD algorithm for determining accurate neighborhood graphs was applied to a biomedical application involving cell segmentation and tracking [2]. Time-lapse phase contrast microscopy of epithelial cells moving in a monolayer sheet

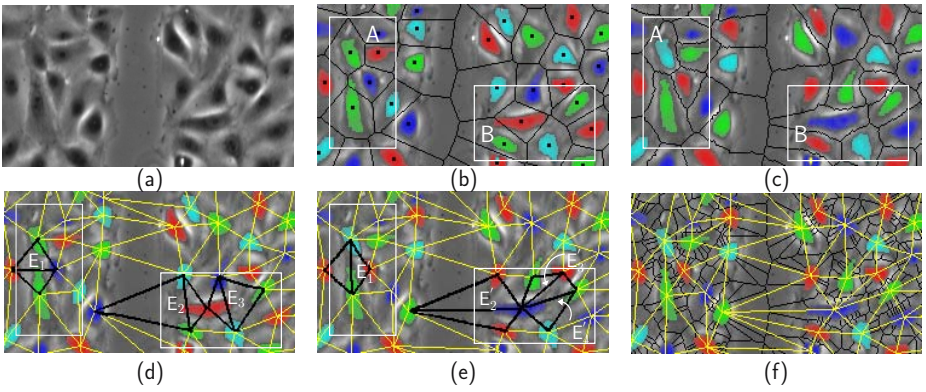

Fig. 5. (a) A representative region from the original image (Frame 46). **(b)** OVD boundaries superimposed with four unique colors. The centroids are represented as small squares. **(c)** GVD boundaries superimposed with four unique colors. **(d)** Neighborhood relationships in two representative regions A and B, when using the OVD from (b). The OVD leads to incorrect neighborhood relationships, as shown by edges E_1, E_2 and E_3. This leads to neighboring cells being assigned the same color during graph-vertex coloring [2]. **(e)** Neighborhood relationships using GVD with correct assignments indicated by $\mathsf{E}_1^{'}$, $\mathsf{E}_2^{'}$ and $\mathsf{E}_3^{'}$ with an additional neighborhood relationship $\mathsf{E}_4^{'}$ that is detected when. **(f)** Neighborhood relationships without pruning branches of the GVD does not affect neighborhood relationships between cells. This feature will be addressed in a different paper. Parameters used in computing the Hamilton-Jacobi GVD are: $\sigma = 0.5$, 9-tap Laplacian kernel with a center weight of 8, $T_{LD} = 0.0$, and $T_{HS} = 5$.

are imaged at $0.13\mu m$ resolution, and appear as a clustering of dark colored nuclei with indistinct boundaries (Fig. 5(a)) [3, 2, 17].

The OVD regions, and associated Delaunay graph based on centroids of cell nuclei in Fig. 5(a) are shown Figs. 5(b) and (d), respectively. Edges E_1, E_2, and E_3 show incorrect object adjacency relationships based on OVD regions A, and B. The object colors are based on graph-vertex coloring and used to implement a fast 4-color level set-based cell segmentation algorithm incorporating spatial coupling constraints [2]. The main feature of the 4-color level set algorithm is to assign different colors to neighboring cells, in order to prevent false merges. From Fig. 5(b) it can be observed from region A that the two green-colored cells are neighbors of each other, yet they are not marked as neighbors when using an OVD. However, in Fig. 5(c) and 5(e), these cells are correctly classified as neighbors when using our proposed GVD algorithm (the cells have been recolored).

The neighborhood adjacency graphs for the synthetic images shown in Figs. 1(a) and 1(d) using the OVD and GVD are shown in Tables 1 and 2, respectively. It is clearly evident that the Hamilton-Jacobi GVD algorithm correctly identifies neighbors of objects in both images, while errors are evident when using OVD to compute the spatial adjacencies of objects. For example, the long thing worm-like object, B_6, is adjacent to smaller elliptical objects B_1, B_3, B_4, B_5, B_7, and, B_8. It does not overlap any other object and has a worm-like influence zone based on the GVD, as seen in Fig. 1(f).

Table 1. Neighborhood map of Fig. 1(a)

$\mathcal{Q}(V_i)$	OVD	GVD
A_1	A_2, A_3	A_2, A_3, A_4
A_2	A_1, A_3	A_1, A_3
A_3	A_1, A_2, A_4, A_5	A_1, A_2, A_4, A_5
A_4	A_3, A_5	A_1, A_3, A_5
A_5	A_3, A_4	A_3, A_4

Table 2. Neighborhood map of Fig. 1(d)

$\mathcal{Q}(V_i)$	OVD	GVD
B_1	B_2, B_3	B_2, B_3, B_6
B_2	B_1, B_3, B_4, B_5	B_1, B_3, B_4, B_5
B_3	B_1, B_2, B_4, B_6	B_1, B_2, B_4, B_6
B_4	B_2, B_3, B_5, B_6	B_2, B_3, B_5, B_6
B_5	B_2, B_4, B_6, B_8	B_2, B_4, B_6
B_6	B_4, B_5, B_7, B_8	B_1, B_3, B_4, B_5, B_7, B_8
B_7	B_6, B_8	B_6, B_8
B_8	B_5, B_6, B_7	B_6, B_7

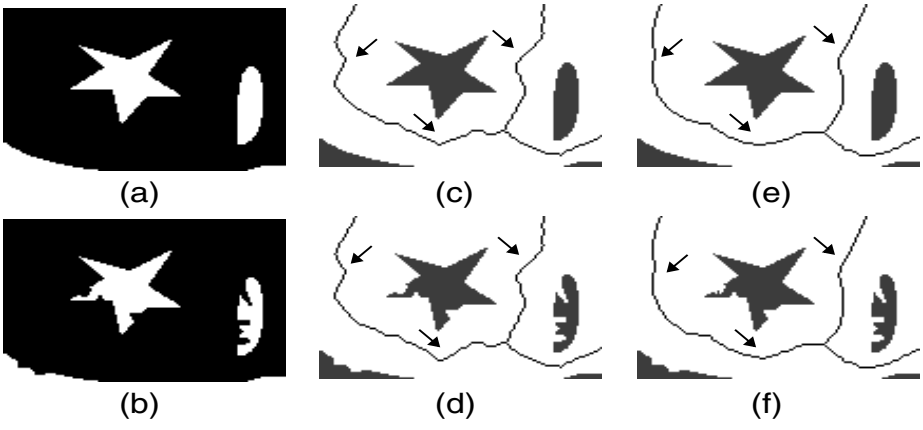

Fig. 6. [**a**]: Subset of objects from Fig. 1(a) showing objects A_4 and A_5. [**b**]: Same objects with perturbed boundaries. [**c** and **d**]: SKIZ-based implementation of GVD with 8-connected label propagation. [**e** and **f**]: Hamilton-Jacobi GVD.

We compare the robustness of the Hamilton-Jacobi GVD algorithm with a watershed-based, fast implementation of SKIZ [18, pg. 170-173] in MATLAB. Figs. 6 (c) and (d), and Figs. 6 (e) and (f) show that the SKIZ-based GVD, and the Hamilton-Jacobi GVD are both relatively insensitive to perturbations in object boundaries as indicated by the arrows. However, Hamilton-Jacobi GVD boundaries are more accurate (same arrows), since the exact EDT is used.

4 Conclusion

In this paper, we have presented a novel algorithm for computing Hamilton-Jacobi based GVD's to build accurate spatial neighborhood adjacency graphs for arbitrarily-shaped objects. Our algorithm extends the Hamilton-Jacobi skeletonization algorithm of Siddiqi *et al.* [12], and is coupled with morphological-based operators to remove spurious regions from the initial GVD boundaries. A fast Laplacian-of-Gaussian (LoG)

filter is used to detect potential GVD boundary locations (i.e., shock points). Useful features of the LoG filter, like the guarantee of closed contours, continuity of ridges, and non-formation of new ridges with an increase in scale (smoothing) makes it appealing for our algorithm. We compare the performance of our Hamilton-Jacobi GVD algorithm, with a previously developed OVD framework for cell segmentation in [2] on real biological, as well as synthetic images. In all instances, we demonstrate the superiority of our GVD algorithm.

As a future work, we would like to present a comparison of our algorithm with other state-of-the-art algorithms described in the literature. Due to the separable nature of the FH-EDT algorithm [13], we can obtain neighborhood relationships between objects in higher dimensions. Hence, we would like to extend our algorithm to $\mathbb{R}^d,\ d > 2$.

References

1. Okabe, A., Boots, B., Sugihara, K., Chiu, S.N.: Spatial Tesselations: Concepts and Applications of Voronoi Diagrams. 2nd edn. John Wiley & Sons Ltd. West Sussex, UK (2000)
2. Nath, S., Palaniappan, K., Bunyak, F.: Cell segmentation using coupled level sets and graph-vertex coloring. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) MICCAI 2006. LNCS, vol. 4190, pp. 101–108. Springer, Heidelberg (2006)
3. Nath, S., Bunyak, F., Palaniappan, K.: Robust tracking of migrating cells using four-color level set segmentation. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 4179–4920. Springer, Heidelberg (2006)
4. Keenan, S., Diamond, J., McCluggage, W., Bharucha, H., Thompson, D., Bartels, P., Hamilton, P.: An automated machine vision system for the histological grading of cervical intraepithelial neoplasia (CIN). J. Pathol. 192, 351–362 (2000)
5. Vincent, L., Masters, B.: Morphological image processing and network analysis of cornea endothelial cell images. In: Proc. SPIE - Image Algebra Morph. Image Proc. III, San Diego, CA, vol. 1769 pp. 212–226 (1992)
6. Choset, H., Walker, S., Ard, K., Burdick, J.: Sensor-based exploration: Incremental construction of the hierarchical generalized voronoi graph. I. Journ. Robotics Res. 19, 126–148 (2000)
7. Aurenhammer, F.: Voronoi diagrams - A survey of a fundamental geometric data structure. ACM Comp. Surveys 23, 345–405 (1991)
8. Sugihara, K.: Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams. Com. Vis. Graph. Image Process 55, 522–531 (1993)
9. Vincent, L.: Graphs and mathematical morphology. Sig. Proc. 16, 365–388 (1989)
10. Lu, Y., Tan, C.: Constructing area Voronoi diagram in document images. In: Proc. 88^{th} IEEE Int. Conf. Doc. Anal. Recog. IEEE Comp. Soc. pp. 342–346 (2005)
11. Hoff III, K., Keyser, J., Lin, M., Manocha, D., Culver, T.: Fast computation of generalized voronoi diagrams using graphics hardware. In: SIGGRAPH-99. 26th Ann. Conf. Comp. Graphics Inter. Tech, pp. 277–286. ACM Press/Addison-Wesley Publications, New York (1999)
12. Siddiqi, K., Bouix, S., Tannenbaum, A., Zucker, S.: Hamilton-Jacobi skeletons. Int. J. Comput. Vis. 48, 215–231 (2002)
13. Felzenswalb, P., Huttenlocher, D.: Distance transforms of sampled functions. Technical Report TR2004-1963. Dept. of Comp. Sci. Cornell University, Ithaca, NY (2004)

14. Rosenfeld, A.: A characterization of parallel thinning algorithms. Inform. Control 29, 286–291 (1975)
15. Cychosz, J.: Efficient binary image thinning using neighborhood maps. In: Graphics gems IV, pp. 465–473. Academic Press Professional, Inc, San Diego, CA, USA (1994)
16. Lam, L., Lee, S., Suen, C.Y.: Thinning methodologies-A comprehensive survey. IEEE Trans. Patt. Anal. Machine Intel. 14, 869–885 (1992)
17. Bunyak, F., Palaniappan, K., Nath, S.K., Baskin, T.I., Dong, G.: Quantitive cell motility for in vitro wound healing using level set-based active contour tracking. In: Proc. 3rd IEEE Int. Symp. Biomed. Imaging (ISBI), Arlington, VA, pp. 1040–1043. IEEE Computer Society Press, Los Alamitos (2006)
18. P.Soille: Morphological Image Analysis. 2 edn. Springer, New York, USA, ISBN 3-540-429883-0 (2004)