Voronoi-Based Segmentation of Cells on Image Manifolds

Thouis R. Jones[1], Anne Carpenter[2], and Polina Golland[1]

[1] MIT CSAIL, Cambridge, MA, USA
[2] Whitehead Institute for Biomedical Research, Cambridge, MA, USA

Abstract. We present a method for finding the boundaries between adjacent regions in an image, where “seed” areas have already been identified in the individual regions to be segmented. This method was motivated by the problem of finding the borders of cells in microscopy images, given a labelling of the nuclei in the images. The method finds the Voronoi region of each seed on a manifold with a metric controlled by local image properties. We discuss similarities to other methods based on image-controlled metrics, such as Geodesic Active Contours, and give a fast algorithm for computing the Voronoi regions. We validate our method against hand-traced boundaries for cell images.

1 Introduction

Image cytometry, the measurement of cell properties from microscopy images, has become an important tool for biological research. In particular, high-throughput experiments rely on automatic processing of images to deal with the large amount of data they produce [3]. A fundamental operation in cell-image analysis is identifying individual cell boundaries. This is often difficult because there are many different staining protocols, leading to dramatically different appearances for cells. Moreover, the difference between cell interior and cell border may not be very pronounced (see Figures 1 and 2). Identification of individual cells allows much more powerful analysis of the resulting data than methods that provide only mean measurements for cell populations. For example, expression data from a protein interaction chip cannot differentiate a bimodal population and a unimodal population if they have the same mean expression levels. Measurements of individual cells prevents conflation of subpopulations.

It is almost always the case that the nuclei of cells are more easily identifiable, because they have a more uniform appearance and shape, are brighter relative to the background when stained, and do not abut one another, as cells do. They are also usually interior to the cells. This leads us to phrase the problem of segmenting cells as one of identifying boundaries between regions given “seeds” in individual regions from which to start the segmentation. Current methods for identifying cells for image cytometry sometimes use a fixed offset around the nuclei. However, this fixed offset requires tuning to different cell types, and does not provide information about phenotypes that cause changes in cell size or shape [2].

Another common approach is to use watershed segmentation to identify cell boundaries [1, 8], however, this is often fragile. Watershed segmentation treats the image as a height field, and segments pixels according to which minimum a drop of water would flow to if placed on that pixel in the height field. Morphological operations are used to impose a limited set of minima, equivalent to our seed regions. Watershed is quite unstable, because a single noisy pixel can allow large groups of pixels to change segmentation, by creating a gap leading to a different minimum. We avoid this fragility in two ways: first, by comparing neighborhoods of pixels rather than individual pixels, and second, by including a regularization factor to provide reasonable behavior when the image data does not contain a strong enough edge between two seed regions. In the limit, our regularization approaches a 2D Voronoi segmentation, i.e., pixels are assigned to the nearest seed region measured in the image plane, without reference to image features.

Our approach is to define a metric in the image plane and to calculate distances from seed regions according to this metric. Pixels are then assigned to cells according to their distance from the corresponding nucleus under that metric. The metric uses information about image edges, both their strength and their orientation, as well as a regularization term corresponding to inter-pixel distance within the image.

2 Method

Our method operates by computing a discretized approximation of the Voronoi regions of each seed on a manifold with a metric controlled by local image features. The metric defines the incremental distance in a particular direction in the image/manifold. Its behavior is such that adjacent pixels with similar surrounds are close to one another, while pixels whose surrounds differ are treated as farther apart.

We introduce a Riemannian metric defined in terms of the image $\mathcal{I}$ and a regularization parameter λ, as

$$\mathbf{G} = \frac{\nabla \mathbf{g}(\mathcal{I}) \nabla \mathbf{g}^T(\mathcal{I}) + \lambda \mathbf{I}}{1 + \lambda}, \tag{1}$$

where $\mathbf{I}$ is the 2×2 identity matrix. The function $\mathbf{g}$ maps images to images, and in our application is generally a small-radius blur. The effect of this blur is to combine a weighted neighborhood in the gradient computation, to avoid relying too much on single (noisy) pixel values. Infinitesimal distances under $\mathbf{G}$ are measured by

$$||d\mathbf{x}||^2_{\mathbf{G}} \equiv d\mathbf{x}^T \mathbf{G} d\mathbf{x} = \frac{(d\mathbf{x}^T \nabla \mathbf{g}(\mathcal{I}))^2 + \lambda (d\mathbf{x}^T d\mathbf{x})^2}{\lambda + 1}. \tag{2}$$

The first term in the numerator of (2), $||d\mathbf{x}^T \nabla \mathbf{g}(\mathcal{I})||^2$, increases distances measured parallel to large gradients in $\mathbf{g}(\mathcal{I})$. The regularization effect of λ can be seen by

$$\lim_{\lambda \to \infty} ||d\mathbf{x}||^2_{\mathbf{G}} = d\mathbf{x}^T d\mathbf{x} = ||d\mathbf{x}||^2_2, \tag{3}$$

i.e., $\mathbf{G}$ becomes more Euclidean as λ increases.

Given (2), we can compute the distance between any two points in an image as the shortest path between those points. We use a discretized approximation, i.e., a chamfer distance, applied to an 8-connected neighborhood. This approximation to the distance is generally no worse than 10% of the correct value [7], which for our application is generally not more than a single pixel.

Pixels overlapping seed regions are initialized to be distance 0 from their seed, and the distances of the remaining pixels are computed by Dijkstra's algorithm [6]. Each pixel is labelled with the seed it is closest to, i.e., the Voronoi region of that seed in the manifold defined by (2). Computing the seed-to-pixel distances in this manner also makes it trivial to limit the segmentation to a predetermined foreground region (for example, the result from a global thresholding step).

The computation of Voronoi regions places the boundary between two adjacent regions at pixels that are equidistant from each seed, as measured on the manifold from (2). The inter-pixel distance is larger where the image is changing more according to $\nabla \mathbf{g}(\mathcal{I})$, so boundaries tend to align with image differences. The regularizing parameter λ allows the user to make the manifold more "flat" according to prior knowledge about region shapes, and the choice of $\mathbf{g}$ (e.g., the radius of smoothing) controls how sensitive distances are to small features. In this work, we use a narrow (3- or 5-pixel radius) Gaussian blur for $\mathbf{g}$.

2.1 Connection to Geodesic Active Contours

Our algorithm is related to Geodesic Active Contours [5]. The full details of that work are not given here, but we discuss the connection briefly.

Active contours can be seen as finding a shortest path in a Riemannian space, where distances between pixels are defined by an edge stopping function $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$.[3] Examining equation (8) from Caselles *et al.* [5] helps establish the similarity:

$$\text{Min} \int_0^1 g(|\nabla \mathcal{I}(\mathcal{C}(q))|)|\mathcal{C}'(q)|dq \tag{4}$$

where $\mathcal{I}$ is the image, $\mathcal{C}(q)$ is the curve on image that we are minimizing over, and q is the parameter along the curve. The edge stopping function g is strictly decreasing and positive, with $g(\infty) = 0$. The effect of g's interaction with $\nabla \mathcal{I}$ is such that the minimum curve follows larger gradients in the image.

The minimization can also be written as (equation (12) of [5])

$$\text{Min} \int_0^{\mathcal{L}(\mathcal{C})} g(|\nabla \mathcal{I}(\mathcal{C}(s)|)ds \tag{5}$$

where s is the arclength parameter for $\mathcal{C}$, and $\mathcal{L}(\mathcal{C})$ is the length of $\mathcal{C}$. Therefore, active contours can be seen as seeking a minimum length curve where the length depends on image characteristics [5].

Our goal is different from active contours, since we hope to find boundaries between regions corresponding to different seeds. However, we do seek shortest paths with a distance metric controlled by image characteristics. We do not wish to follow boundaries, as in active contours, but rather to avoid them (equivalent to making boundaries "far"

[3] In this subsection, g is an edge stopping function, not the same as $\mathbf{g}$ in the previous subsection.

from the seed regions in our formulation). However, just as Voronoi regions can be defined in terms of shortest paths, our segmentation algorithm can be defined in terms of shortest paths defined by equation (1) and image properties. The metrics in the two approaches differ, specifically in their treatment of edges in the image: edges in the active contour setting have a metric that makes points along the edge closer, while our metric makes the points across the edge more separated. Moreover, active contours use a directionally uniform metric, while ours is not, since it is larger across edges rather than along them. However, the goal is the same: allow the computation of inter-pixel distances that simplify the problem at hand and map it to a simpler framework. In both cases, the problem reduces to that of finding shortest paths (though this is simply a jumping-off point for active contours to more powerful and efficient methods such as level-sets).

3 Experiments and Evaluation

In order to provide insight into the behavior of the metric defined in (1), we experiment with synthetic data and adjust λ, trading off the effect of image features on the region segmentation versus the Euclidean distance from the seed. As can be seen in Figures 3 and 4, the segmentation approaches the Voronoi diagram of the seed regions (i.e., ignoring image features) as λ increases. Also, note that in the noisy example less regularization (i.e., a lower value for λ) is needed to achieve similar segmentations because the noise makes the edges in the image less pronounced than in the noise-free example. In effect, increasing noise provides a form of regularization.

Our algorithm is currently part of an automatic image cytometry program [4], and has been used in a variety of experiments with several cell types of varying morphologies. To evaluate our algorithm on real world data, we compare it to the manual segmentation of cell images. Sixteen images taken from Drosophila cells stained for DNA (to label nuclei) and Actin (a cytoskeletal protein, to show the cell body) were outlined. First, nuclei were outlined by hand. The nuclear outlines were overlaid on the cell images, and one cell per nucleus was outlined. Our algorithm was then applied to the same cell images and nuclear-outlines data, and the results compared by computing the signed distance between boundary pixels. It effectively computes the distance for each pixel in the automatic segmentation boundary to the corresponding manual segmentation boundary. Hand-outlined nuclei were used, rather than automatically segmented nuclei, since our algorithm does not address nuclei segmentation, and we want the comparison to be as meaningful as possible. In general, automatic segmentation of nuclei is fairly simple (See section 1.)

Our algorithm does not compute a foreground/background separation, but instead relies on a such a label for each pixel to be given as input. For comparing our algorithm to the manual segmentation, we compute the foreground pixels as the union of cells identified in the manual segmentation. Methods exist for automatically choosing a foreground/background labelling, but distinguishing foreground from background is not part of our algorithm.

For the purposes of this comparison, we also use seed regions from hand-outlined nuclei. In general, nuclei are more compact, separated, and brightly stained than cells,

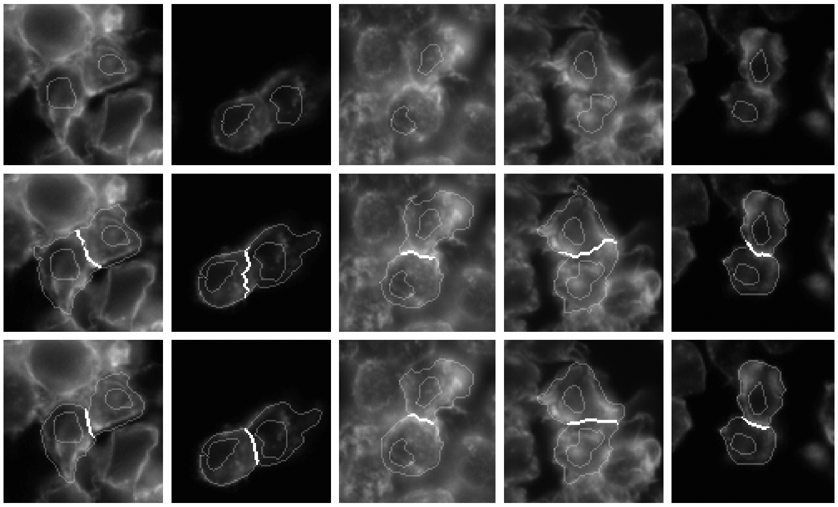

Fig. 1. Typical results from our algorithm. Internal shapes are seed regions, taken from manual segmentation of nuclei, and are the same in both rows. Bold lines show cell/cell boundaries that are compared. Top row: Cell images, with nuclei outlined. Middle row: Automatic segmentation with our method. Bottom row: Manual segmentation.

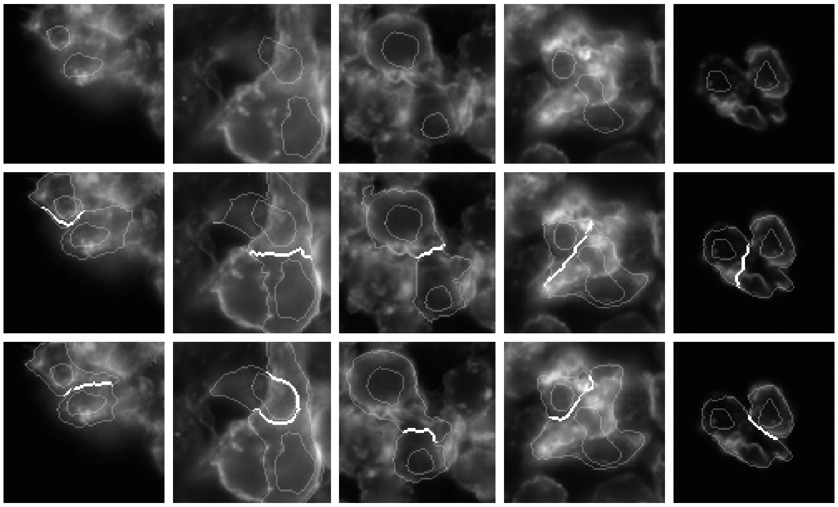

Fig. 2. Five worst outliers from evaluation set, as measured by distance between manual and automatic segmentation boundaries. Internal shapes are seed regions, taken from manual segmentation of nuclei, and are the same in both rows. Bold lines show cell/cell boundaries that are compared. Top row: Cell images, with nuclei outlined. Middle row: Automatic segmentation with our method. Bottom row: Manual segmentation.

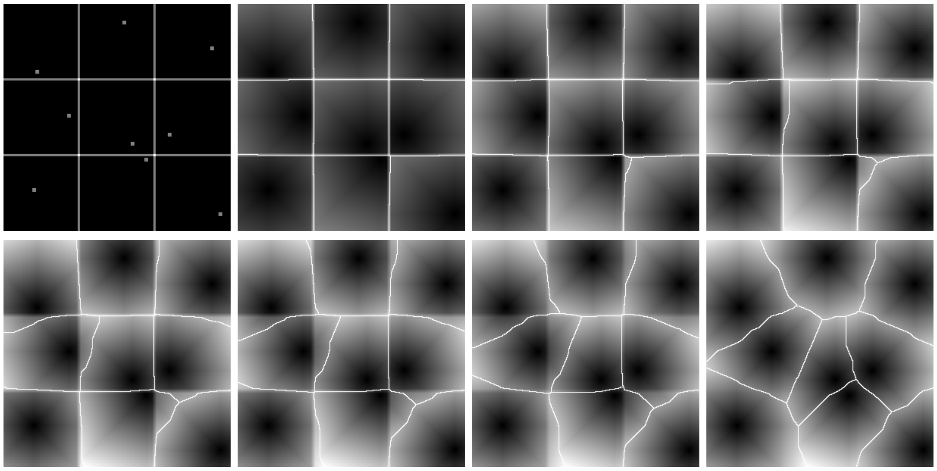

Fig. 3. Synthetic example. The input image is in the upper left, with seed sites marked with dots. From left to right across the two rows, the resulting distances calculated with our metric are shown, with the resulting segmentation overlaid white lines, for λ equal to 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, and 3.0. The segmentation lines follow the ridges in the distances function. As can be seen, as λ increases, the segmentation approaches the Voronoi diagram of the seed regions.

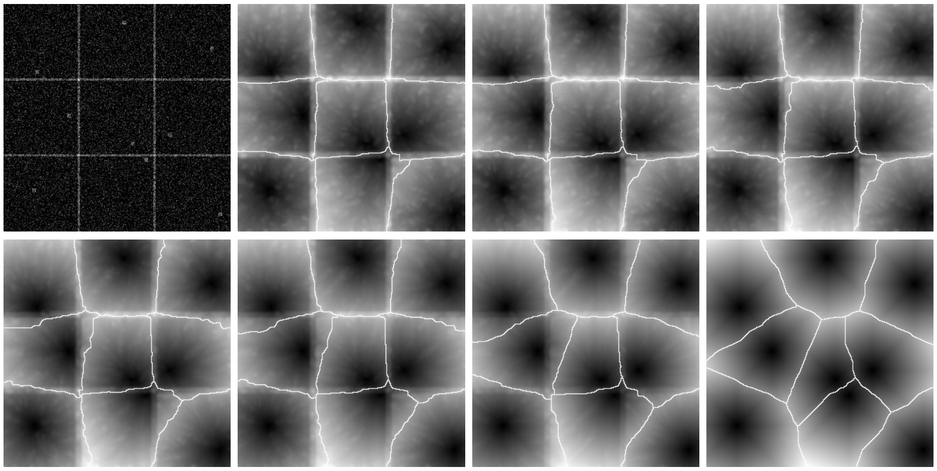

Fig. 4. Synthetic example with noise. The same input as in Figure 3 is used, with zero-mean Gaussian noise with standard deviation 0.5 added to each pixel. Edges were 1.0 and background was 0.0 before noise was added. The layout is the same as in Figure 3, but with λ equal to 0.025, 0.05, 0.075, 0.1, 0.125, 0.2, and 0.75.

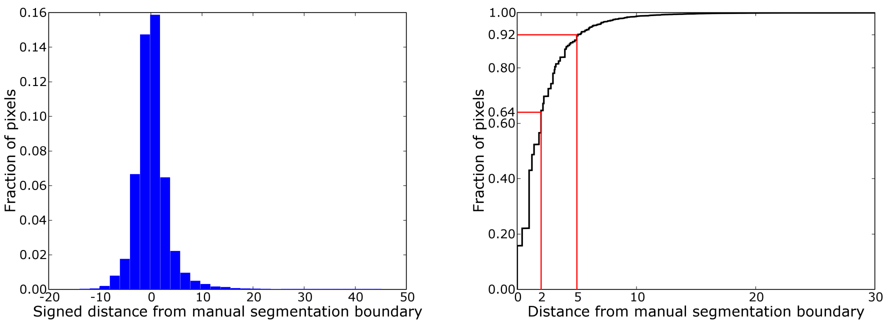

Fig. 5. Combined histogram for the signed distances and cumulative distribution of absolute distances from automatic segmentation to manual segmentation for the sixteen images in our test set.

and so are usually easily segmented by simple thresholding. We chose to use the nuclei from manual segmentation in order to keep the conditions for manual segmentation and our algorithm as similar as possible.

Under these conditions, our algorithm is only responsible for computing cell-cell boundaries, rather than cell-background boundaries, which are fixed. We therefore only consider the distance between boundary pixels in the two images that have a common cell-label on either border. When evaluating the algorithm, we use the one-sided signed distance (negative inside cells) from the automatic segmentation to the manual segmentation. We set λ in (1) to 0.05 times the distance between the average foreground and background pixel intensities on a per-image basis. This value for λ was found to be close to optimal in our experiments, with fairly stable behavior for a reasonably large range (within a factor of two). Our test set includes a wide variety of cell types, with different sizes and morphologies. In general, most applications would have more homogeneous data, to which λ should be tuned.

Sixteen images made up the test set. Each image was roughly 512x512 pixels on a side, with cells roughly 25 pixels in diameter, and 80 cells per image on average. Across the entire set, there were 21.6k pixels on a cell-cell boundary in the automatic segmentation. A histogram of their signed distances with respect to the manual segmentation is shown in Figure 5. Sixty-four percent (14.0k) of the boundary pixels in the automatic segmentation are within 2 pixels in distance from the corresponding manual boundary. Ninety-two percent (19.8k) of the pixels are within 5 pixels. The accuracy of the hand-labelling is around 3 pixels, based on the width of the marker used to outline the cells.

Typical results on the test set are shown in Figure 1. We show the top 5 worst outliers from the data set, based on maximum boundary distance, in Figure 2. In some cases, the automatic method has "chosen" a different edge in the image to use as the cell-cell boundary. In others, the close proximity of the nucleus to a cell boundary has caused the automatic segmentation to move past the boundary chosen in the manual segmentation, causing large deviations between the two.

4 Discussion

In some segmentation tasks seed regions for segmentation are easily identified. Nuclei act as such seed regions in our application, the segmentation of cells in microscopy images. The more difficult task is finding the boundaries between cells that share a common border. Cell appearance in the images is not uniform, varying depending on the type of cells and the protocol used to stain them, and simplistic approaches are not sufficiently accurate or robust.

We have demonstrated an algorithm for segmentation of image regions based on seed areas that respects image boundaries as defined by a difference operator. The algorithm computes an approximation of the Voronoi region of a seed on a manifold, implicitly defined by a difference operator that operates on image neighborhoods. The only assumption that the algorithm relies on is that images change more near the borders of regions. This is similar to the behavior of Geodesic Active Contours, with an inversion in the behavior of the metrics near edges. We approximate distances using a chamfer-like difference operator, and use Dijkstra's algorithm for computing individual regions quickly.

Our algorithm is currently implemented in an automatic image cytometry package [4], and has been used successfully in several experiments with a variety of cell types and morphologies. We have compared our algorithm to manual segmentation by an expert of microscopy cell images. Our algorithm performs well. There are a few failure cases, perhaps due to the fairly simple prior on cell shape and size that our metric implies. A more complex prior could incorporate measures of, e.g., cell roundness or cell area. Although our method is more accurate for segmenting cells than others, such as using fixed offsets from nuclei or the watershed transform, the true test of its usefulness is in whether it produces more accurate measurements of cellular phenotypes. In order to understand how segmentation accuracy affects measurement accuracy, we plan to validate against data from flow cytometry or another method not based on image processing.

In the future, we would like to explore other choices for $\mathbf{g}$ in (1). It might also be possible to use the distance defined by our metric to compute foreground/background labellings and to detect poorly-segmented cells.

Acknowledgements

The authors would like to thank In-Han Kang for his work on validation of the algorithm, and Chris Gang for the hand-outlined nucleus and cell images.

References

1. S. Beucher "The Watershed Transformation Applied to Image Segmentation," *Scanning Microscopy International*, Vol. 6, pp.299–314, 1992.

2. Boutros M., A. Kiger, S. Armknecht, K. Kerr, M. Hild, B. Koch, S. Haas, Heidelberg Fly Array Consortium, R. Paro and N. Perrimon "Genome-wide RNAi Analysis of Cell Growth and Viability in Drosophila," *Science*, Vol. 303, pp. 832–835, 2004.
3. A.E. Carpenter and D.M. Sabatini "Systematic Genome-wide Screens of Gene Function," *Nature Reviews Genetics*, Vol. 5, pp. 11–22, 2004.
4. "CellProfiler: versatile software for high throughput image analysis," A.E. Carpenter, T.R. Jones, D.B. Wheeler, D.A. Guertin, I.H. Kang, P. Golland, and D.M. Sabatini, *in preparation.*
5. V. Caselles, R. Kimmel, and G. Sapiro "Geodesic Active Contours," *Int. J. Comput. Vision*, Vol. 22, pp. 61–79, 1997.
6. E. W. Dijkstra "A Note on Two Problems in Connection with Graphs," *Numerische Math.*, Vol. 1, pp. 269–271, 1959.
7. U. Montanari "A Method for Obtaining Skeletons Using a Quasi-Euclidean Distance," *Journal of the ACM*, Vol. 15, pp. 600-624, 1968.
8. C. Wählby "Algorithms for Applied Digital Image Cytometry," PhD Thesis, 2003