Probabilistic Segmentation and Analysis of Horizontal Cells

Vebjorn Ljosa and Ambuj K. Singh
University of California, Santa Barbara
{ljosa,ambuj}@cs.ucsb.edu

Abstract

Because images of neurons show interweaved processes from multiple cells, it is hard to determine which pixels belong to each cell, and consequently to analyze the images automatically. To manage these difficulties, we introduce probabilistic segmentation*, in which each pixel is assigned a probability of belonging to each cell instead of being categorically assigned to one cell. We propose a randomized algorithm for probabilistic segmentation. The algorithm is based on repeated, intensity-weighted random walks on the image, and leads to improved segmentation quality.*

Analysis and mining techniques can utilize the more nuanced and complete information that the probabilistic segmentation yields about an image. Such techniques can then compute probabilistic values, which indicate the level of confidence that can be placed in them.

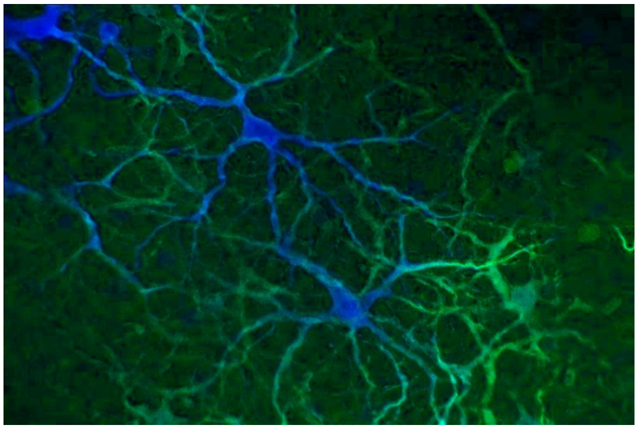

Figure 1. Confocal micrograph of three horizontal cells in a detached cat retina, labeled by anti-neurofilament (green) and anti-calbindin (blue).

1 Introduction

The role of microscopy, a cornerstone of many fields of biology, is changing. Rather than a likeness subject to visual inspection, the micrograph is becoming a quantitative measurement subject to formal analysis. Together with high-throughput acquisition techniques, this change in mind set may bring data-driven research, and the success it has had in genetics, to other fields.

Neuroscience is one field that could benefit greatly from data-driven research. Databases for images are starting to become available [10, 12, 13], but quantitative analyses are still far from being central to the research. One reason for this is that neurons *in vitro* (in a dish) develop and respond to injuries and other stimuli quite differently than neurons *in vivo* (in tissue). One must therefore image and study cells in tissue—a much more difficult undertaking, both in terms of acquisition and analysis.

One approach to understanding the vast complexity of the brain is to study the retina, which is part of the central nervous system, but is accessible and contains relatively few classes of cells organized in well-defined layers. The electrical signals generated by the photoreceptors in response to light go through the bipolar cells and ganglion cells before they are finally passed through the optic nerve into the rest of the brain. There are other types of neurons in the retina in addition to these three, however. In particular, *horizontal cells* are fairly flat neurons located in the outer plexiform layer, where the photoreceptors and bipolar cells meet [7]. As their name indicates, they provide connections that are horizontal, i.e., perpendicular to the main direction of signaling. Figure 1 shows three horizontal cells. (All images have had their brightness and contrast adjusted so the morphology of the cells would be visible in print.)

Images like the one in Figure 1 are acquired by staining the retina with fluorescent antibodies to proteins that are present in (and specific to) the cells or parts of cells of interest. Two, three, or even four antibodies can be used together as long as they fluoresce in response to different wavelengths of light. The tissue in Figure 1 was stained with anti-neurofilament (green) and anti-calbindin (blue), which together label entire horizontal cells in cat retinas.

The processes extending from the cell bodies are called *neurites*, and generally taper as they extend farther from the cell. The intensity in the image also decreases. This is a result of the cell becoming thinner in the third dimension:

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

there is less cytoplasm, and therefore fewer proteins and fluorescent antibodies. It is hard to decide which pixels belong to which cell because the neurites are intertwined with neurites from the other two cells and with neurites from cells the bodies of which are outside the field of view.

Several kinds of questions are of interest:

1. Morphological mining. Are there patterns in the number of neurites or how they branch? Do they taper as they extend away from the cell body? Do they grow longer or branch more in response to injury?

2. Protein distribution mining. Does the distribution of a protein within the cells follow certain patterns? Why do some neurites express much neurofilament and little calbindin or vice versa?

3. Neuron connection mining. Are connections between neurons related to changes in morphology or protein distribution? Do the connections between neurons change in response to injury?

One would like to know how the answers to these questions differ between experimental conditions (treatments, etc.) and between different species.

The problem is not so much that such analyses are tedious to perform manually, but that they are nearly impossible for humans to perform reliably. The fundamental reason is that extracting the features that form the basis for the analyses is inherently error prone. As a simple example, the bottom cell in Figure 1 has at least four neurites, but it may have as many as eight, depending on whether some neurites belong to this cell or to another cell. The uncertainty can sometimes be controlled—in the simple case of cell counting, this is done by stringent protocols for how to count, blinding, and multiple human counters—but this is infeasible in most cases. Yet, the quantitative studies that have been done have had strong impact; see for instance Rex et al.'s work [11] on protein expression in photoreceptors.

This paper therefore addresses probabilistic segmentation and analysis of images. It will become clear that even though we work with horizontal cells in this paper, our approach is applicable not only to neurons elsewhere in the brain but also to many other kinds of images. We have previously [1] mined biomedical images in the aggregate, extracting visual features from chunks of tissue containing numerous cells and mining them to construct a vocabulary of latent concepts. In a sense the present paper attacks the problem from the opposite angle, identifying and measuring single cells and their properties in order to cast light on what the aggregate analysis cannot.

We propose, in Section 2, the first probabilistic segmentation algorithm, by which we mean that the algorithm computes not only a set of pixels that seem to belong to a certain cell, but also states how confident it is that each pixel belongs to the cell. Section 3 evaluates the algorithm experimentally. Finally, in Section 4, we describe how the probabilistic segmentation result can be used to automatically measure morphological properties of the cell—a first step in answering the questions listed above.

2 A probabilistic segmentation algorithm

Our algorithm is motivated by a simple diffusion model of the flow of newly synthesized proteins in the cell. Consider a hypothetical protein, which is produced near the center of the cell body, then distributed throughout the cell by diffusion. The cell body is nearly round, so the beginning of the diffusion process is a symmetric, Brownian motion, but assuming that the protein is not secreted through the cell membrane, further diffusion is limited to places that are part of the cell, i.e., the neurites. We simulate this model by a discrete random walk, starting in the center of the cell.

An immediate problem for the simulation is that the image does not tell us directly where the cell membrane is. However, the boundary of the cell can for the most part be inferred from the gradient of the distributions of the protein(s) being imaged by observing where the intensity in the image drops off sharply.

The random walk consists of a large number of steps. Each step is to one of the eight pixels neighboring the current location in the image, chosen at random, but not uniformly: The decision is biased by the relative intensity of the neighbors so that the step is more likely to be in the direction of a bright neighbor pixel. A separate matrix of the same dimensions as the image is updated at each step to keep count of how many times each pixel has been visited. Our conclusions about which pixels belong to the cell with what probability will be based on this "visit record" matrix.

The heuristic of using the gradient of the intensity to keep the walk within the cell works well when the cell is surrounded by unlabeled background, but fails when a neurite touches that of another cell. Such a "bridge" between two cells is usually a thin, dim, peripheral part of a neurite, so it is quite unlikely that the walk will stray across it, but for a long walk this is bound to happen at some point—with disastrous consequences for the algorithm described so far, for it is unlikely that the walk will cross back into the original cell, so it will keep visiting pixels in the wrong cell.

The problem can be solved by modifying the algorithm to perform a *repeated* random walk. After each step an unfair coin is tossed, and with a small probability c the walk returns to its original starting point before continuing. This solves the problem of crossing into other cells because a walk that strays into the wrong cell will soon return to the original starting point. Because most walks will never cross the bridge, the original cell will be visited much more than the other.

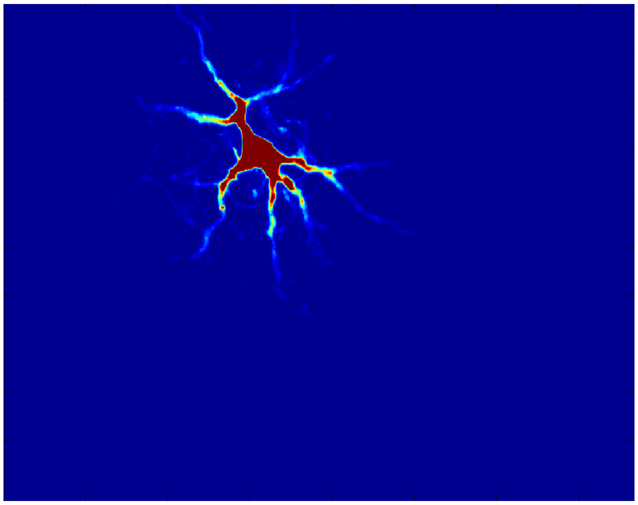

Figure 2. One of the cells in Figure 1, successfully segmented by the repeated random walk algorithm.

It may appear that even crossing the bridge once and walking around in the wrong cell for a little while is a misfortune. However, this behavior is actually a blessing in disguise. The reason is that a connection between neurites from different cells can look quite similar to a branching point in a neurite—so much so that human experts cannot reliably distinguish them. If the algorithm had to make a definite choice between staying in the original cell or crossing into the other, a wrong choice would have huge impact on the segmentation result and consequently on the analysis results. The randomized solution avoids this by visiting the region beyond the bridge proportionally to how well it is connected to the original cell. An analysis algorithm can recognize that there is doubt about the extent of the cell and conclude, for instance, that there is a 0.8 and 0.2 probability that the neurite is 30 µm and 40 µm long, respectively.

Figure 2 shows the segmentation result for one of the cells in Figure 1. The color of each pixel corresponds to an element in the visit record matrix, and indicates how many times that pixel was visited by the repeated random walk. Note that pixels farther from the center are generally visited less. This is consistent with the diffusion model, and makes sense from a probabilistic point of view, for a path from the center to a pixel far away has more opportunities to make mistakes and cross into other cells, so we can be less certain that the pixel actually belongs to the original cell. Notice also that the wider neurites are followed more often—again consistent with what we expect.

The visit record matrix constitutes a graded segmentation result—pixels visited more are more likely to be part of the cell—but how do we convert the visit counts to probabilities? The mapping should be linear, but contrary to intuition, assigning probability 1 to the pixel visited most and probability 0 to pixels not visited at all may not be the best mapping. The reason is twofold. First, there will be some fluorescence even in parts of the image that do not contain the protein intended to be imaged, because of background staining and autofluorescence. To correct for this, we should map visits below a certain threshold to probability 0. Second, the peripheral parts of the cell body receive fewer visits than the central parts, even though we have prior knowledge that the cell body is part of the cell. We can correct for this by mapping visits above a certain threshold to probability 1. (Technically, we are then computing $P(\text{pixel} \in \text{cell} \mid \text{cell body} \in \text{cell})$.) The resulting matrix of probabilities is called a *probabilistic mask*, or *pmask*.

2.1 Formulation as eigenvector problem

Although a simulation-based implementation of the algorithm, as described above, is sufficiently efficient (5 s per cell for a 768-by-512-pixel image), it is interesting to note that the pmask can also be computed by solving an eigenvector problem. Each step of the walk can be written as

$$\boldsymbol{x} := (1-c)\boldsymbol{P}\boldsymbol{x} + c\boldsymbol{s}. \tag{1}$$

Here, $\boldsymbol{x}$ is the pmask, P is a (non-symmetric) transition matrix, c is the restart probability, and s is a vector that indicates the center of the cell (the element corresponding to the pixel at the center of the cell is one, the rest are zero).

If the image is $m \times n$ pixels, then $\boldsymbol{x}$ and $\boldsymbol{s}$ have mn elements, and $\boldsymbol{P}$ has mn rows and mn columns.

Çamoğlu et al. [4] show that Eq. (1) converges to the stationary probability distribution of the Markov chain with transition matrix $\boldsymbol{Q} = \{Q_{ij}\}$, defined by

$$Q_{ij} = \begin{cases} (1-c)P_{ij} & \text{if } s_i \neq 1 \\ (1-c)P_{ij} + c & \text{if } s_i = 1. \end{cases} \tag{2}$$

At convergence, $\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{x}$. Because Q is column-normalized, its largest eigenvalue is 1, so $\boldsymbol{x}$ is the corresponding eigenvector. The eigenvector problem can be solved quickly because $\boldsymbol{P}$ and $\boldsymbol{Q}$ are very sparse: each row has only eight nonzero elements, corresponding to the possible next steps.

3 Experimental evaluation

We evaluate our segmentation algorithm experimentally by comparing it to the "seeded" (or "marker-based") watershed algorithm [14], the state of the art for this kind of segmentation problems.

The watershed solution converts the image into a landscape by lowering a pixel of intensity a to a depth a units below the ground. The brightest parts of the image thus become the deepest valleys. The landscape is then modified so that local minima occur only at the center of each cell (the

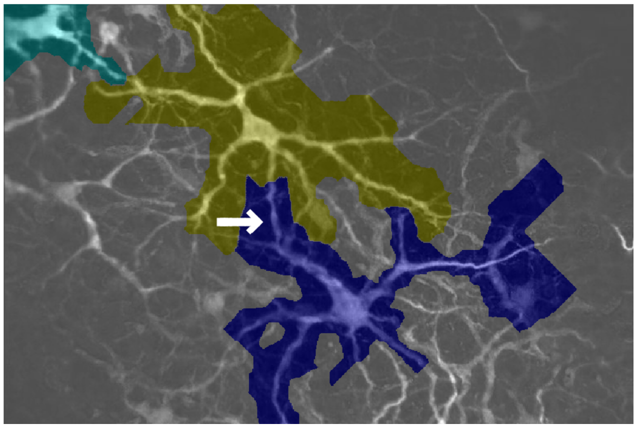

Figure 3. The seeded watershed algorithm's segmentation result, superimposed on the original image.

foreground markers), and local maxima occur only in areas known to be outside the cell (the background markers). The background markers were constructed by thresholding followed by careful closing and dilation. The size of the structuring elements for the closing and dilation was critical, and had to be selected by trial and error; with the wrong values, the algorithm would make huge mistakes. Note that absent an automatic method for finding background markers, the watershed algorithm is not a practical solution. In contrast, the foreground markers, which are also used as starting points for our repeated random walks, can be found automatically [2].

Figure 3 shows the result of the seeded watershed algorithm on the image in Figure 1. The colored areas indicate which pixels the algorithm assigns to each of the three cells. The original image is shown in the background as a frame of reference. We would not expect the algorithm to produce a perfect segmentation, but it is striking that when it does make mistakes, they are large mistakes: Long pieces of neurites are completely missing from the lower cell, and most of the neurite indicated by the arrow is misclassified as belonging to the lower cell when it actually belongs to the cell above it.

The watershed segmentation, as shown, is not very specific: Large areas of the background are assigned to the cells. This can be remedied, however, by assuming that pixels in the original image with intensity below a certain threshold (i.e., low concentration of the protein being imaged) are not part of the cell. By increasing the threshold, we can improve specificity at the expense of sensitivity. The repeated random walk algorithm has a similar tradeoff: Modifying the mapping from visit count to probability, as discussed in Section 2, affects specificity and sensitivity.

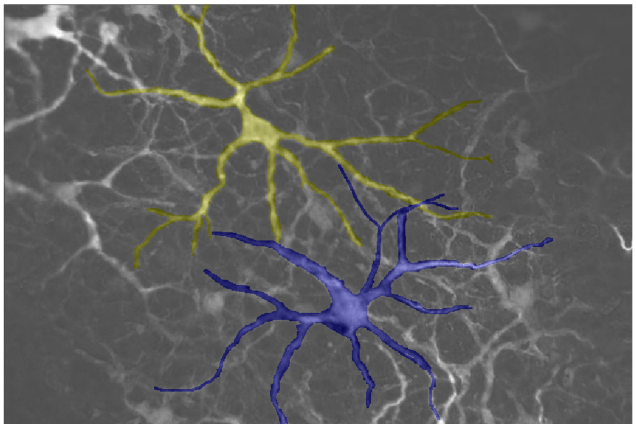

Figure 4. Segmentation ground truth.

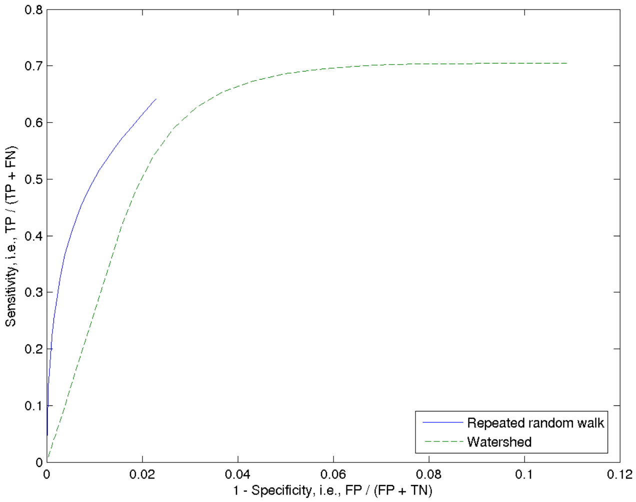

Figure 5. The repeated random walk algorithm dominates seeded watershed in terms of receiver operating characteristics (ROC).

In order to quantify the specificity and sensitivity of the algorithms, we segmented two cells manually, as shown in Figure 4. Even a human cannot say for certain which neurites and branches belong to the cell, so we deliberately erred on the side of including too much in the manually segmented cells, as this leads to conservative measurements of the algorithms' sensitivity.

When the random walk algorithm assigned a probability p to a pixel, we added $(1-p)$ to the count of false negatives and p to the count of true positives if the pixel was truly part of the cell. Similarly, we added p to the count of false positives and $(1-p)$ to the count of true negatives if the pixel was not truly part of the cell.

Figure 5 plots receiver operating characteristics (ROC) for the two algorithms. An ideal algorithm would have a curve that touches the top left corner of the space (perfect sensitivity and specificity). We see that the repeated random walk dominates the seeded watershed.

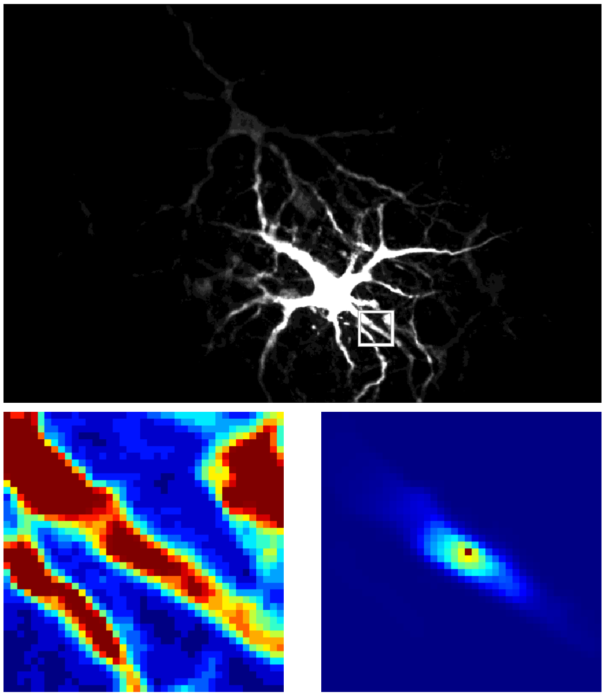

Figure 6. We find the direction of the neurite by PCA on a region around the point (top). PCA fails to detect the direction of the neurite when other neurites are in close proximity (left). However, a second repeated random walk identifies the neurite (right).

4 Probabilistic analyses

The probabilistic nature of the segmentation result provides extra information that analysis techniques can exploit in order to produce a more useful end result. Because of space limitations, we only consider a simple morphological measurement: the thickness of a neurite at a given point Q. This is only a first step in analyzing the image, but gives a flavor of how analysis and mining techniques must be adapted to handle probabilistic values.

We would like to measure the thickness along a line that is perpendicular to the neurite at Q. The first challenge is to find this line. The local direction of the neurite can be found by principal component analysis (PCA) [9]. Restricting our attention to a small region R around Q, we find the coordinates of all pixels in R that have an above-average probability of being in the cell (according to the pmask). In most cases, the first principal component of these points gives the direction of the neurite at Q.

This method fails, however, when there are other neurites in close enough proximity to intersect R. Figure 6 shows an example. The neurite we want to measure extends from the top-left to the bottom-right corner, but interfering neurites in the other two corners cause the direction of largest variance to be nearly perpendicular to the neurite. In order to avoid this problem, we apply the probabilistic segmentation method from Section 2 a second time in order to find the parts of the region that are most likely to be connected to Q by tissue. The right panel of Figure 6 shows that the method is effective at detecting the neurite of interest. PCA can now easily find the direction of the neurite.

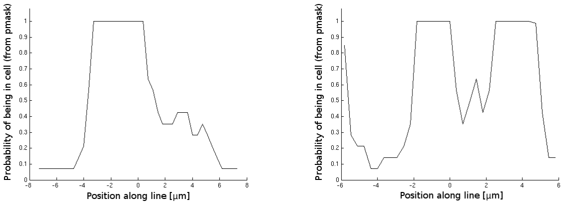

Figure 7. For two neurites, probabilities (from the pmask) for each pixel along a line across the neurite.

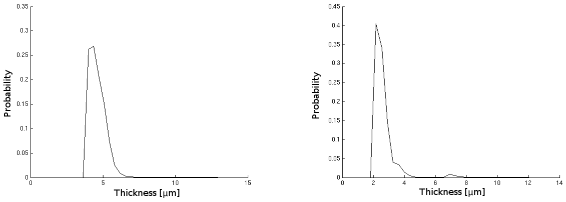

Figure 8. The resulting probabilistic values for the thickness of the two neurites.

Once we have found the line perpendicular to the neurite, we can read off the pmask values along the line, resulting in a graph like the ones in Figure 7. Even equipped with this graph, it is difficult to determine the thickness of the neurite—either because the sides of the neurites are not very well defined, as in the left panel of Figure 7, or because it is not clear whether there is one neurite or two neurites next to each other, as in the right panel. We circumvent the problem of choosing a threshold by computing $P(w)$, the probability that the neurite is w pixels thick, for $w \in \{1,2,3,\ldots\}$. The neurite is w pixels thick if, along the line, there is a sequence of $w+2$ pixels such that the first and last are not part of the cell but the remaining w pixels are. If p_i is the probability (from the pmask) that the i-th pixel in this sequence is part of the cell, the probability that a neurite of thickness w starts at pixel s is

$$P(w,s) = (1-p_{s-1}) \prod_{i=s}^{s+w-1} p_i (1-p_{s+w}). \qquad (3)$$

We can compute $P(w)$ by trying all relevant start positions:

$$P(w) = \max_{s=q-w+1}^{q} P(w,s) \qquad (4)$$

Only start positions from $q-w+1$ to q are tried because other start positions would only find other neurites than the one that Q is part of.

Figure 8 plots the $P(w)$ functions that correspond to the pmask values in Figure 7. The function $P(w)$ has the same shape as the probability density function (pdf) of the neurite's thickness (although the values on the second axis are scaled as a result of the discrete fashion in which $P(w)$ is obtained), and we say that the measurement of w is a *probabilistic value*. The probabilistic value is directly useful to biologists because it conveys the variability in the measurement. The right panel of Figure 8 indicates, for instance, that the neurite is most likely between 2 μm and 4 μm thick, but there is also a small possibility that it is around 7 μm thick. The probabilistic value is even more useful in automatic analysis and mining because it allows the techniques to deal intelligently with measurements of different accuracy. Probabilistic values have received much recent attention in the database community because they are useful not only for databases of images and other scientific data, but also for databases of moving objects, sensor readings, and historical business transactions. This work has mostly focused on building index structures that allow for efficient search by the probabilistic values [3, 5, 6, 8].

5 Conclusion

Microscopy is the main source of data for many biological and medical disciplines. In many kinds of micrographs, each pixel is a *quantitative* measurement—e.g., of the concentration of a certain protein at a location in the tissue. This opens the door for analysis and mining algorithms that produce probabilistic values instead of a definite best guess.

We have proposed a novel algorithm for segmenting horizontal cells and other objects with intertwined processes. The algorithm assigns to each pixel a probability that the pixel is part of the cell. We have further shown how this segmentation can serve as a basis for automated analysis of the morphological properties of cells.

Future work will apply the algorithm to three-dimensional confocal micrographs, explore variations of the random walk that avoid sharp turns, and develop mining algorithms that can work on probabilistic measurements derived from the pmask.

Together, probabilistic segmentation and analyses that return probabilistic values provide a foundation for giving quantitative answers to the questions of neuroscience—and a bridgehead for the data mining community into the world of microscopy-based biology.

Acknowledgements. We would like to thank Geoffrey P. Lewis from the laboratory of Steven K. Fisher at UCSB for providing the retinal micrographs. This work was supported in part by grant no. ITR-0331697 from the National Science Foundation.

References

[1] A. Bhattacharya, V. Ljosa, J.-Y. Pan, M. R. Verardo, H. Yang, C. Faloutsos, and A. K. Singh. ViVo: Visual vocabulary construction for mining biomedical images. In *Proc. ICDM*, pages 50–57, 2005.

[2] J. Byun, N. Vu, B. Sumengen, and B. Manjunath. Quantitative analysis of immunofluorescent retinal images. In *Proc. Int. Symp. Biomed. Imaging*, 2006.

[3] C. Böhm, A. Pryakhin, and M. Schubert. The Gauss-tree: Efficient object identification in databases of probabilistic feature vectors. In *Proc. ICDE*, 2006.

[4] O. Çamoğlu, T. Can, and A. K. Singh. Integrating multi-attribute similarity networks for robust rrepresentation of the protein space. *Bioinformatics*, 22(13):1585–1592, 2006. (Proof in supplement.).

[5] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. *TKDE*, 16(9):1112–1127, Sept. 2004.

[6] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. S. Vitter. Efficient indexing methods for probabilistic threshold queries over uncertain data. In *Proc. VLDB*, 2004.

[7] J. E. Dowling. *The Retina: An Approachable Part of the Brain*. Belknap, 1987.

[8] A. Faradjian, J. Gehrke, and P. Bonnet. GADT: A probability space ADT for representing and querying the physical world. In *Proc. ICDE*, 2002.

[9] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[10] M. Martone, S. Zhang, A. Gupta, X. Qian, H. He, D. Price, M. W. M, S. Santini, and M. Ellisman. The Cell-Centered Database: A database for multiscale structural and protein localization data from light and electron microscopy. *Neuroinformatics*, 1(3), 2003.

[11] T. S. Rex, R. N. Fariss, G. P. Lewis, K. A. Linberg, I. Sokal, and S. K. Fisher. A survey of molecular expression by photoreceptors after experimental retinal detachment. *Invest. Opthal. Vis. Sci.*, 43(4), 2002.

[12] A. K. Singh, B. Manjunath, and R. F. Murphy. A distributed database for bio-molecular images. *SIGMOD Record*, 33(2):65–71, 2004.

[13] J. R. Swedlow, I. Goldberg, E. Brauner, and P. K. Sorger. Informatics and quantitative analysis in biological imaging. *Science*, 300:100–102, 2003.

[14] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. PAMI*, 13:583–598, 1991.