IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis, San Francisco, CA, June 14, 2010

Structural Correspondence as A Contour Grouping Problem

Elena Bernardis
University of Pennsylvania
Philadelphia, PA 19104

Stella X. Yu
Boston College
Chestnut Hill, MA 02467

Abstract

We present a novel viewpoint which approaches the structural correspondence across an image stack in the 3D space as solving a contour grouping problem. Finding 3D cellular tubes becomes finding closed contours. We derive grouping cues between cells in adjacent slices based on their ability to relate in the 3D space. Those that form a long 3D tube in the space become the most salient contour, while those of shorter lengths become less salient. In the spectral graph-theoretical framework for contour grouping, such a separation by the contour length is reflected in complex eigenvectors of different magnitudes, from which these 3D tubes of varying lengths can thus be extracted, obviating the need for identifying missing correspondences.

1. Introduction

Extracting 3D tubular cell structures across a stack of 2D image slices (Fig. 1) requires establishing cellular correspondences between images (Fig. 2), and we approach it as a contour grouping problem.

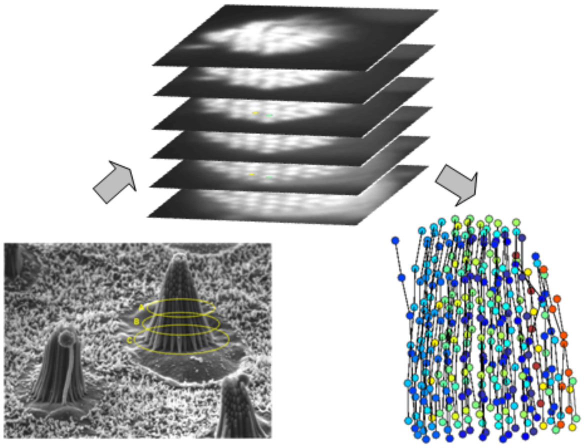

Figure 1. 3D Stereocilia segmentation. Hair cells are composed of tens of stereocilia organized in an organ-pipe-like formation of increasing height. We propose to solve the correspondence between cells across the 2D image stack as a contour grouping problem.

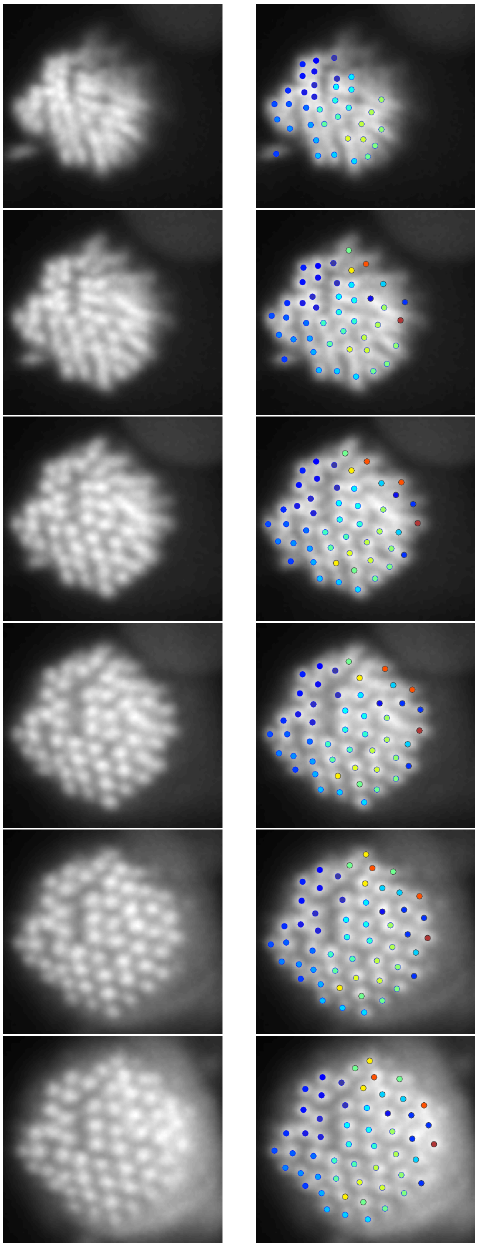

Figure 2. Extracting 3D tubes by finding correspondences (coded in color) across the image stack. Missing correspondences simply result in shorter contours in our contour grouping framework.

The most straightforward approach is to consider all the images in a stack simultaneously and solve a 3D pixel segmentation problem. There are many 3D image segmentation methods, e.g. level sets [1], clustering algorithms [2, 3] and region growing [4], etc. While this formulation has an output format that naturally describes 3D tubes, it is unclear how grouping cues within and across individual 2D images can be properly integrated. Another major issue is scalability. The number of pixels increases with the increasing image resolution and number of slices, often rendering the computation infeasible.

An alternative approach is to solve a series of 2D pixel segmentations independently and then combine the results to obtain the 3D structures [5, 6, 7, 8]. Since cells of different tubes often look alike, there are few good features to distinguish them. In practice, it is problematic to identify and cluster 3D tubes of varying lengths.

Both approaches need to address the structural correspondence problem, which is explicit in the 2D segmentation approach and implicit in the 3D segmentation, e.g. in the derivation of motion cues linking the deformation of one image slice to the next image in the stack.

We propose to solve the structural correspondence between slices in the 3D space by solving a contour grouping problem. Here the *contours* are imaginary closed contour cycles cutting through the stack, going through the centers of the regions that have correspondences in the 3D space.

We derive grouping cues between cells in adjacent slices based on their ability to relate in the 3D space. Those that form a long 3D tube in the space become the most salient contour, while those of shorter lengths become less salient.

We solve the contour grouping problem in the spectral graph-theoretical framework [9]. The separation by the contour length is reflected in complex eigenvectors of different magnitudes, from which these 3D tubes of various lengths can be extracted.

The most appealing strength of our formulation is that missing correspondences no longer poses a special and difficult problem. They simply lead to shorter contours which are extracted from cycles of shorter lengths.

2. Contour Grouping for 3D Correspondence

To find the 3D tubes transversing a 2D image stack, we formulate the cellular correspondence as a contour grouping problem. Each 3D tube is represented by a contour starting from the cell in the lower stack, transversing the stack and turning backward once the end of the hair cell is reached, returning back to the initial slice.

Our method is inspired by the *Untangling Cycles* [9] model for contour grouping, which extracts contours by searching for salient cycles of a random walk, within a spectral graph partitioning framework. In our setup, nodes of the weighted graph no longer represent edge pixels of possible 2D contours on an image, instead, they represent segment centers from each individual stack.

2.1. Untangling Cycles for Contour Grouping

Within a single image, modeling contour grouping as a spectral graph partitioning problem can be seen as finding persistent cycles in a random walk along a weighted graph $G(V, E)$, where weights W correspond to 'edge' nodes V, derived from an initial edge map, and edges E between nodes are given by spatial proximity in the original image. We generate a directed random walk matrix

$$\overrightarrow{P} = D^{-1}W \tag{1}$$

normalized with respect to the outgoing connections from each node, i.e. D is a diagonal matrix with entries

$$D(i,i) = \sum_j [W(i,j)] \tag{2}$$

The criterion is more robust to contour leakages.

The idea behind *Untangling Cycles*, is that paths within the random walk have to return to their initial starting point (to guarantee closeness of the contour) and have to do so in the same period t (to guarantee the repeated transversing of a salient contour rather than an accidental returning from surrounding clutter). In this context, salient contours can be thought of as 1D cycles, with a special returning pattern quantified through a *peakness* measure $R(i, T)$ of the random walk probability pattern at steps of multiples of T.

The key observation [9] is that $R(i, T)$ closely relates to complex eigenvalues of the transition matrix P, instead of real eigenvalues, showing that it is the complex eigenvalues with proper phase angle and magnitude that lead to repeated peaks. A random walk step on a 1D cycle tends to stay within the cycle, while moving a fixed amount forward in the cyclic ordering. Therefore, the link between finding image contours and distinguish them from clutter amounts to searching for *persistent cycles* in this random walk.

2.2. Graph Setup for Structural Correspondence

We start with stack of images and their respective segmentations and assign to each segment a node. With this new set of vertices, nodes can now have the same image spatial coordinates, while belonging to a different stack number. Edges between nodes are added if the corresponding segments are lying in adjacent images, while, within the same stack, only a self returning edge is added at each node. The two frameworks are sketched in Fig. 3. The weight associated to each edge has two components.

First, we recall the single image scenario, where each node had an associated direction to it, and a 'good' contour was one maximizing smoothness. We still seek smoothness in terms of the 3D tubular structure, so that intuitively

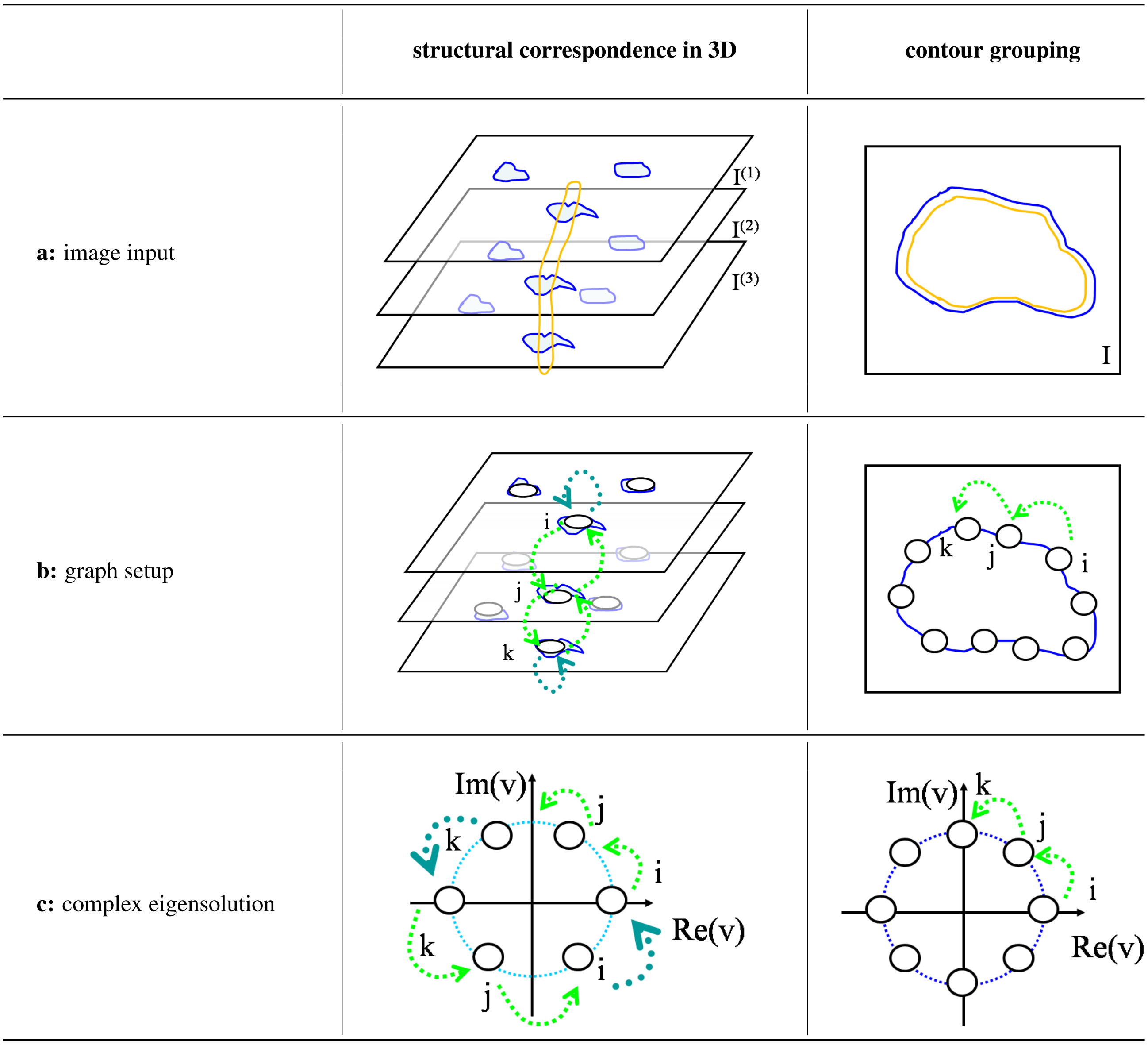

Figure 3. Structural correspondence as a contour grouping problem. We contrast our method with the traditional Untangling Cycles model. In order to solve structure correspondence and find the cycles (**a**, yellow), nodes in the graph (**b**) are no longer edge pixels; instead, they represent segments on the individual images. In order to have close contours we add a return link to each outer stack node (dark green). Cycles through the 3D tubular structures then correspond to cycles in the complex eigenvectors v of a random walk matrix (**c**).

each tube does not bend too much while transversing the stacks. If we imagine each node with an arrow pointing downwards, measuring bending between two nodes can be simplified by projecting them onto one single plane and measuring the spatial distance between them. Letting the positions of node i and j be d_i and d_j respectively on the $x-y$ planes of the individual images, we define

$$\xi(i,j) = \exp\left(-|d_j - d_i|/\sigma\right) \tag{3}$$

We have empirically found that the introduction of a complex component θ to the weights allows better cycle discrimination in the embedding space. ψ encodes the number steps taken in the random walk, so that jumping between stacks, in terms of the n images stacks $\{I^{(1)}, I^{(2)}, \ldots, I^{(n)}\}$, can be written as

$$\psi_{i \to j} = t - s, \qquad i \in I^{(s)}, j \in I^{(t)} \qquad s,t \in 1,\ldots,n \tag{4}$$

where the unit step is given by $\psi = 1$. The final weight

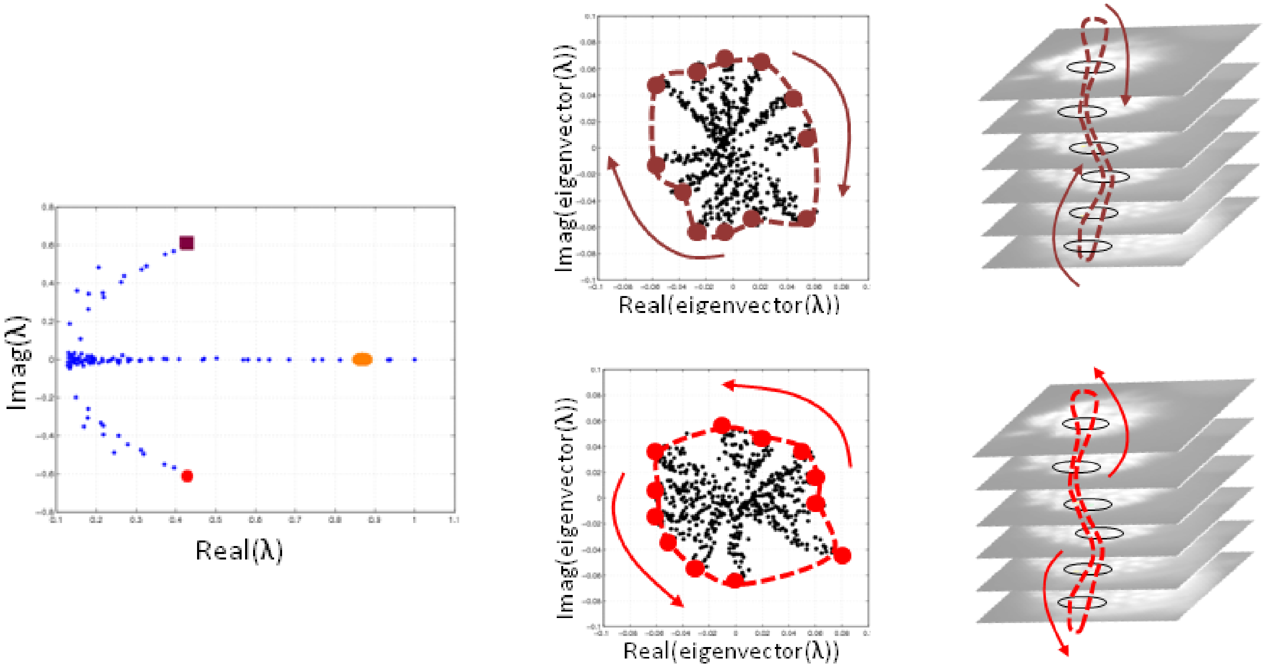

Figure 4. Embedding Space. While real eigenvalues (orange ellipse) collapse eigenvectors on the real line, eigenvectors corresponding to complex eigenvalues with large phase angle and real components (purple square, red circle) encode cycles information: each cycle corresponds to a cycle in the original graph, which itself encodes a cycle though the image stacks. Positive and negative phase angles encode clock and anti-clockwise cycles respectively.

between two nodes will then be given by:

$$w_{i\to j} = \begin{cases} \xi(i,j) + (\psi_{i\to j})\mathtt{i} & i \neq j \\ \xi(i,i) + \mathtt{i} & i = j = 1, n \\ \xi(i,i) * 0.1 + \mathtt{i} & i = j = 2, \dots, n-1 \end{cases} \qquad (5)$$

In order to have close contours we add a return link to each node Fig. 3. Adding a returning edge at each layer guarantees that cycles of shorter lengths will also be able to be picked up in the random walk, hence dealing with missing correspondence throughout the stacks.

2.3. Circular Embedding for Random Walk Cycles

In order to understand the intuition behind the eigenvectors of the random walk matrix P, [9] showed that the eigenvectors are an approximate solution to an ideal cost for circular embedding of salient contour grouping.

A circular embedding (Fig. 4) is a mapping between the vertex set V of the original graph to a circle plus the origin: $\mathcal{O}_{circ} : V \mapsto (r, \theta) : \mathcal{O}_{circ}(i) = x_i = (r_i, \theta_i)$, where r_i is the circle radius which can only take a positive fixed value r_0 or 0. θ_i is the angle associated with each node. The ideal embedding encodes both the *cut* and the *ordering* of graph nodes, and maximizes the following score:

$$C_e(r, \theta, \Delta\theta) = \sum_{\substack{\theta_i < \theta_j \leq \theta_i + 2\Delta\theta \\ r_i > 0,\ r_j > 0}} \overrightarrow{P}_{ij}/|S| \cdot \frac{1}{\Delta\theta}, \qquad (6)$$

where S is a subset of graph nodes and $\Delta\theta = \overline{\theta_j - \theta_i}$ is the *average jumping angle*.

Setting $u = x$, $v = u \cdot e^{-i\Delta\theta}$, and $c = t_0 e^{-i\Delta\theta}$, we can rewrite Eqn. (6) as maximizing $\mathtt{Real}((u^H \overrightarrow{P} v \cdot c)/(u^H v))$ and it is equivalent to the following:

$$\max_{u,v \in \mathbb{C}^n} \mathtt{Real}(u^H \overrightarrow{P} v) \qquad \text{s.t. } u^H v = c. \qquad (7)$$

which leads exactly to $\overrightarrow{P}$'s complex eigenvectors.

2.4. Algorithm

Given an image stack with their associated segments,

1. Build a graph $G(V, E)$ where the nodes V correspond to the segments throughout the stack.
2. Compute the weight matrix by Eqn. 5.
3. Solve the complex eigenvectors of the associated random walk matrix $\overrightarrow{P}$.
4. Extract contours from cycles in the embedding space [9].

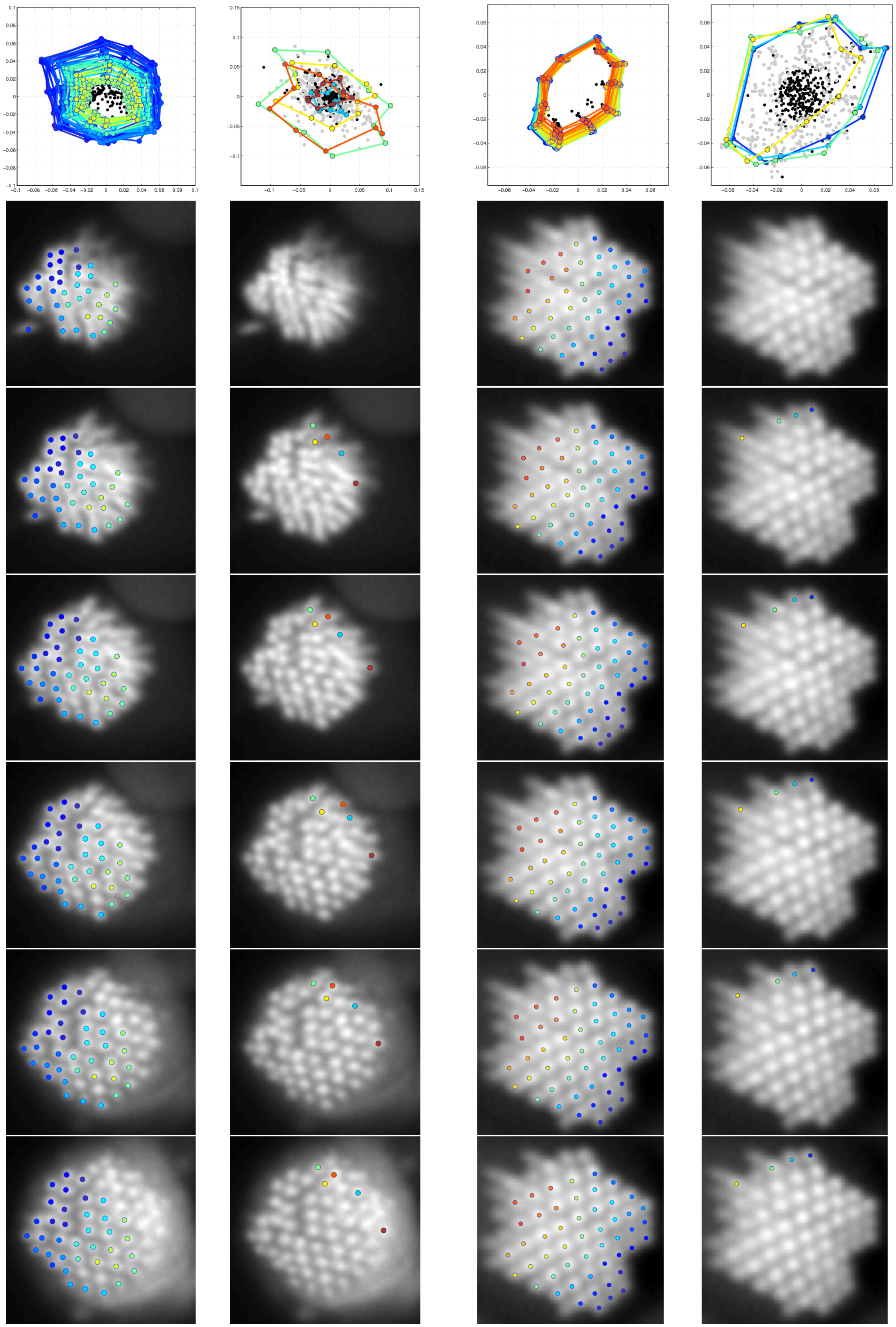

Figure 5. Sample results illustrating 3D tubes shrinking (left 2 columns) and shifting (right 2 columns) across the stack. The first row shows two eigenvectors and extracted contour cycles of lengths 6 and 5. Each cycle in the embedding space is color-coded to show the 3D cell correspondences throughout the 6-image stack. Longer cycles are first extracted from the eigenvector.

3. Experiments

We have 20 stacks of microscopic haircell images. For each stack, we selected 6 image slices. The parameters used were kept constant for all stacks: connectivity radius of 25 pixels and $\sigma = 5$. The choice of radius relates to the amount of shifting allowed between the images. Cells centers were found using the 2D segmentation method of [10].

We allowed cells to connect only up to two slices forward. Cylindrical tubes representing hair cells in 3D are then picked up individually as cycles in the eigenvectors Fig. 4. Given n stacks, cycles of length 6 represent cells that can be seen throughout the slices, while cycles of length $5, 4, \ldots$ represent the ones with missing correspondences, i.e. disappearing cells in the upper stacks.

Our implementation in MATLAB takes about 1 second to find the correspondences in a stack of about 60 tubes.

We show results for two stacks in Fig. 5, illustrating the main two challenges present in these type of cells. Hair cell bundles are formed by an organ-pipe like structure of tubular cells, with the radially outer ones of shorter lengths when compared to the central ones. Depending on each hair bundle and imaging technique, the difference in shrinking of the cells can be seen either as cells shifting in space, possibly assuming ellipsoidal cross-sections before dissipating in the next image layer, or as cells concentrically shrinking, if the images and the hair bundle actually align with respect to its center. Fig. 5 illustrates how two different eigenvectors for each image stack contain the information of cycles of different length.

4. Summary

We present a contour grouping approach to extract tubular structures across image stacks. The key insight is to view the structural correspondence problem as finding closed contours across the image stack.

We formulate it in the spectral graph partitioning framework, where the random walk matrix is constructed from complex graph weights capable of encoding stack ordering. While the resulting eigenvectors correctly encode the tubular structure information for all cells, regardless of their lengths throughout the stacks.

What's most appealing about our method is that cycles found can handle missing correspondences in the form of disappearing, shrinking, and shifting cells. In addition, this particular choice of contour grouping with complex weights allows all salient cycles of the same length to be extracted from the same eigenvector.

Acknowledgements

The authors would like to thank Gang Song at UPenn for helpful comments, and thank Medha Pathak and David Corey at Harvard University for providing haircell images.

This research is funded by NSF CAREER IIS-0644204 and a Clare Boothe Luce Professorship to Stella X. Yu.

References

[1] Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. Journal of Computational Physics **79** (1988) 12–49 2

[2] Coleman, G., Andrews, H.: Image segmentation by clustering. In: Proc. IEEE. (1979) 773–785 2

[3] Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. In: PAMI. Volume 15. (1993) 1101–1113 2

[4] Haralick, R., Shapiro, L.: Image segmentation techniques. In: Computer Vision, Graphics and Image Processing. Volume 29(1). (1985) 100–132 2

[5] Malladi, R., Sethian, J.A.: Level set methods for curvature flow, image enchancement, and shape recovery in medical images. In: Proc. of Conf. on Visualization and Mathematics, Springer-Verlag (1997) 329–45 2

[6] Meyer, F.: Topographic distance and watershed lines. Signal Process. **38** (1994) 113–125 2

[7] Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watersheds: a new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In: ICCV. (2009) 2

[8] Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI **22** (2000) 888–905 2

[9] Zhu, Q., Song, G., Shi, J.: Untangling cycles for contour grouping. In: ICCV. (2007) 2, 4

[10] Bernardis, E., Yu, S.X.: Finding dots: Segmentation as poping out regions from boundaries. In: CVPR. (2010) 6