Cell Motion Analysis Without Explicit Tracking

Richard Souvenir[1], Jerrod Kraftchick[1], Sangho Lee[2], Mark G. Clemens[2], and Min C. Shin[1]
[1]Department of Computer Science, [2]Department of Biology
University of North Carolina at Charlotte
{souvenir,jpkraftc,sahlee,mgclemen,mcshin}@uncc.edu

Abstract

Automated cell tracking using in vivo imagery is difficult, in general, due to the noise inherent in the imaging process, occlusions, varied cell appearance over time, motion of other tissue (distractors), and cells traveling in and out of the image plane. For certain types of cells these problems are exacerbated due to erratic motion patterns. In this paper, we introduce the Radial Flow Transform, which provides motion estimates for objects of interest in a scene without explicitly tracking each object. The transform is robust to misdetected objects, temporally-disjoint motion events, and can represent multiple directions of flow at a single location. We provide operations to convert to and from a vector field representation. This allows for intuitive reasoning about the motion patterns in a scene. We demonstrate results on synthetic data and in vivo microscopy video of a mouse liver.

1. Introduction

A variety of imaging techniques exist for the purpose of viewing the behavior of cells in tissue. The motion patterns of these cells can be important to microcirculation researchers as anomalous patterns may be indicative of insufficient blood flow, blockages, or even the presence of a tumor. Tracking the position of cells over time could provide the type of data that would be useful to answering questions about the patterns of cell motion. Manual tracking is, in the best case, tedious and, in the worst case, impractical due to the high number of cells in a single scene. Automated cell tracking using biomedical imagery is difficult and the results are unreliable due to the noise inherent in the imaging process, occlusions of cells with other cells and tissues, varied cell appearance over time, motion of other tissue (distractors), and cells traveling out of the image plane. Figure 1 shows an example image from *in vivo* microscopy of the surface of a mouse liver. In this image, the brighter locations represent cells and the image is labeled to highlight some of the challenges involved in cell tracking.

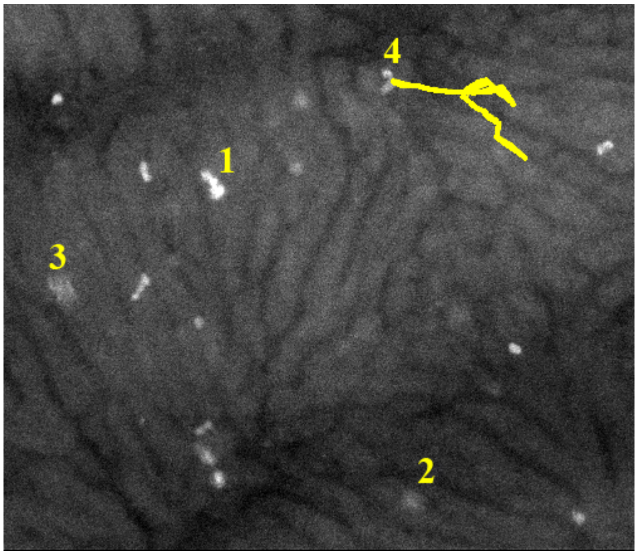


Figure 1. *In vivo* microscopy of a the surface of a mouse liver. The cells appear to fluoresce due to a green fluorescent protein (GFP) filter. This image highlights the challenges involved in tracking cells (specifically NKT cells), namely (1) multiple cell occlusion, (2) tissue motion (distractors), (3) cells out of the imaging plane, and (4) erratic motion patterns.

In this paper, we aim to provide estimates of motion at each location in the field of view, without explicitly tracking individual cells. To ground this work, we focus on the motion analysis of an especially difficult class of immune cells to track: Natural Killer T (NKT) cells. Compared to other types of blood cells, NKT cells are especially difficult to track due to their generally erratic motion patterns.

The literature on object tracking is vast [11], and many approaches have been explored specifically for cell tracking [1, 7]. Our work has similar goals and can even be used to complement these methods. In this paper, we want to provide estimates of motion at each location in the field of view. However, to overcome the problems of tracking depicted in Figure 1, we seek to avoid tracking individual cells.

Previous work considers calculating the flow of objects without (or prior to) explicit tracking. In [2], the authors use

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

artificial neural networks to classify pixels as either vessel or tissue, based on color and temporal variance for vessel segmentation. The authors assume that the direction of flow is parallel to the centerline of the detected vessel and use this constraint to aid in the tracking of cells.

Outside of biomedical image analysis, work related to our approach can be found in road extraction literature, where the goal is to find coherent paths of motion. In [5], the authors use contextual information, such as the detection of vehicles, to detect the position and dominant direction of roads. In [8], the authors model the spatio-temporal image derivatives, and define roads as connected paths of consistent motion. Also, the idea of using histograms of local patches of motion to find motion patterns has been used for gesture recognition [10]. These problem domains, however, differ significantly from cell motion since we cannot assume consistent motion (e.g., cars) nor a global underlying pattern (e.g., gestures).

Unlike the related methods, we do not seek to learn a dominant direction of motion at each location, but wish to provide a construct capable of being queried for desired motions in a region, including complex motion patterns where, unlike roads, the objects do not only travel in a single direction at a given location. We introduce the *Radial Flow Transform (RFT)*, which provides motion estimates for objects of interest (e.g., cells) in a scene without explicitly tracking each object. The transform is robust to misdetected objects, temporally disjoint motion events, and can represent multiple directions of flow at a single location. In Section 2, we describe how the transform is used to calculate flow estimates for detected objects of interest and describe how to convert between visually-intuitive representations of flow and the RFT, for both querying and describing motion patterns in a scene. In Section 3, we demonstrate a specific application to medical image analysis and show how we detect NKT cells using a supervised learning approach. In Section 4 we show results on both artificially-generated and *in vivo* microscopy data. Finally, in Section 5, we conclude and discuss future directions for this project.

2. Radial Flow Transform

The Radial Flow Transform (RFT) was designed to address two problems in motion analysis: (1) detection of specific motion patterns in a scene and (2) visualization of motion in specific regions of a scene. RFT is most useful for scenes where the optical flow is difficult to calculate, and it is possible to detect objects of interest, but difficult to track them due to erratic motion patterns. RFT generalizes the sequential (possibly noisy) detection of objects of interest into flow estimates.

The RFT can be applied to a classification map, $\mathcal{C}$, obtained as the result of a detection process, where $\mathcal{C}_t(\vec{x}) \in [0,1]$ is the detection confidence of some object at pixel location $\vec{x}$ at time t. We model the distribution of the radial flow at a pixel as a function of the direction. So, for each pixel location, $\vec{x}$, we estimate the radial flow $\mathcal{R}(\vec{x}, \theta_i)$ in direction $\theta_i \in \Theta$ where Θ is a set of (generally evenly-spaced) angles. The RFT is designed for situations where the input is noisy and sparse; this binning step, which decreases the granularity of output, allows us to accumulate motion estimates in a more robust way. In our experiments, we use 8 evenly spaced representative directions $(0°, 45°, \textit{etc.})$. Figure 2 shows a close-up visualization of the radial flow for three image patches: (1) mostly left motion, (2) mostly left and right motion, and (3) a "converging from all directions" motion.

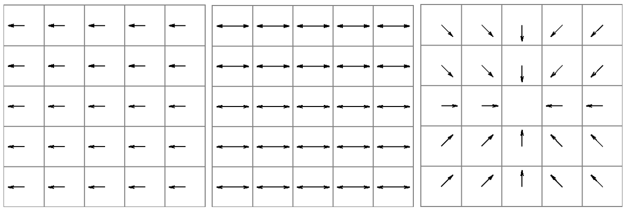


Figure 2. Three examples of the radial flow, $\mathcal{R}^{\theta}(\vec{x})$, for some location, $\vec{x}$. (a) motion to the left, (b) motion both to the left and right and (c) a "converging from all directions" motion. Each cell represents a pixel location and the vectors describe the dominant direction(s) of motion.

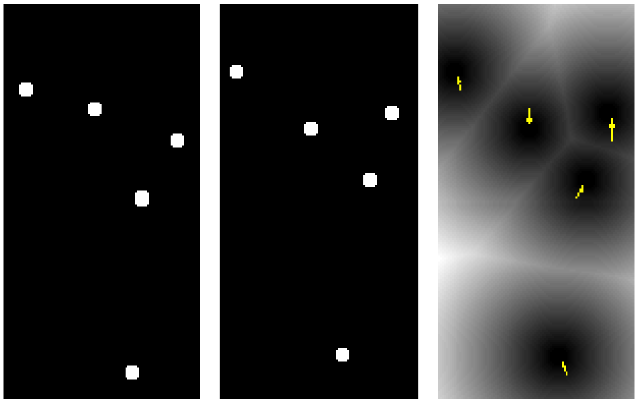


Figure 3. Estimating Flow. This image depicts how flow estimates are calculated for detected objects in sequential frames. The intensity of the image represents the Euclidean distance transform for objects in the current frame. The tail of the arrow represents a detected object in the previous frame and the direction of flow is calculated using the nearest-neighbor assumption.

In this section, we describe how we estimate the flow $F_t(\vec{x})$ at each time, t, and how these values are accumulated to estimate $\mathcal{R}(\vec{x}, \theta_i)$.

2.1. Flow Estimation

Here, we seek to exploit the detection results represented by a confidence map, $\mathcal{C}$, to estimate the visual flow of adjacent frames. Figure 3 depicts the estimate of visual flow for two consecutive frames from a synthetic data set. To estimate the flow, without explicitly tracking each cell, we apply the nearest-neighbor assumption and calculate the distance from each pixel in frame $t-1$ to the nearest positively classified pixel in frame t:

$$D_t(\vec{x}) = \min_{\vec{x}' \in \mathcal{C}_t^+} ||\vec{x} - \vec{x}'||_2, \tag{1}$$

where $\mathcal{C}_t^+$ is the set of locations positively classified at time t. Figure 3 shows the Euclidean distance transform for the positively classified regions in frame t and the locations of the positively classified regions from frame $t-1$ marked at the tail of the arrows, which point in the direction of flow. Given that we are primarily interested in the motion patterns of the objects of interest, we estimate the flow at regions of the image containing positively classified pixels at time $t-1$ and all of the pixels along the path from a classified pixel in frame $t-1$ to the nearest pixel in frame t. For each of these pixels, $\vec{x}$, our estimate of the flow at time t is:

$$F_t(\vec{x}) = \nabla D_t(\vec{x}) \cdot D_t(\vec{x}). \tag{2}$$

2.2. Accumulating Flow Estimates

The flow estimates, F are accumulated to provide a compact representation of the flow at each location in the image:

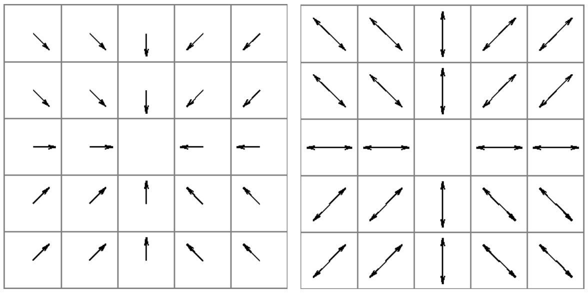
$$\mathcal{R}(\vec{x}, \theta_i) = \sum_t \Psi(F_t(\vec{x}), \Theta, \theta_i)\Phi(\mathcal{C}_t(\vec{x})) \tag{3}$$

where $\Psi(\vec{v}, \Theta, \theta)$ is an indicator function which returns 1 if the dot product between $\vec{v}$ and the unit vector $\langle \cos\theta, \sin\theta \rangle$ is the highest of all the angles in Θ. This function serves to find the correct binning of the motion estimate, F. Φ represents an application-dependent weighting function. There are many choices for Φ, including:

- $\Phi(\mathcal{C}_i(\vec{x})) = 1$, where Φ serves to count the number of instances flow estimates in direction θ.
- $\Phi(\mathcal{C}_i(\vec{x})) \sim \mathcal{C}_i(\vec{x})$, where Φ serves to keep a count, weighted by the classification score, of instances of flow estimates in direction θ.
- $\Phi(\mathcal{C}_i(\vec{x})) \sim |F_i(\vec{x})|^{-1}$, where Φ serves to keep a count, weighted by the reliability of the nearest-neighbor estimate, of instances of flow estimates in direction θ. If tracking results are available, Φ would keep a count based on the weighted reliability of tracking rather than nearest-neighbor estimates.

In our experiments, we have empirically determined that $\Phi(\mathcal{C}_i(\vec{x})) = 1$ produces reasonable results and is robust to noisy flow measurements.

Up to this point, we have only referred to using classification results to derive the flow estimates. However, this method can also be used as a complement to traditional cell tracking methods. If inter-frame tracking results are available, $\mathcal{R}$ is calculated as outlined in Equation 3, except tracking results can be directly substituted for $F_t(\vec{x})$.



(a) Attracting filter (b) Sentry filter

Figure 4. Radial Flow Transform filters for the *attracting* and *sentry* motion patterns.

2.3. RFT Filters

The Radial Flow Transform is useful for queries of motion patterns. This is accomplished using *RFT filters* which are analogous to image filters and applied similarly.

Consider an $l_w * l_h$ image window, which we define as the support of the filter. For each location, x_s in this window, we define $\mathcal{S}_0(x_s, \theta_i) = 1$ if we wish to search for a motion pattern with flow in direction θ_i at position $\vec{x}_s$ and $\mathcal{S}_0(x_s, \theta_i) = 0$ otherwise. The RFT filter is then defined as:

$$\mathcal{S}(x_s, \theta_i) = \mathcal{S}_0(x_s, \theta_i) - \frac{1}{l_w * l_h} \sum_{x_j} \mathcal{S}_0(x_j, \theta_i). \tag{4}$$

RFT filters are then used to search the RFT of an input video to detect flow patterns. Binning the estimates of flow, as described at the start of this section, allows for quickly evaluating the pattern match score $P(\vec{x})$ by performing 3D convolution using the filter and the RFT.

$$P = \mathcal{R} * \mathcal{S} \tag{5}$$

The Radial Flow Transform is designed to query for and examine motion patterns in video. To that end, we provide an intuitive method for the design of RFT filters and for the visualization the RFT. The remaining challenge is to convert between a visually-intuitive representation (vector flow fields) and the RFT-based representation.

2.4. RFT Conversion

One benefit to the RFT is the straight-forward conversion to and from a visually-intuitive vector flow field descriptor. Figure 4 shows two example visualizations of RFT filters. The left image in the figure depicts a motion pattern where the objects are attracted to some location. The right image depicts a motion pattern where the objects are both attracted to and repelled by some location. (Note the multiple direction arrows for a single image location.) Let $V(\vec{x})$ represent a set of vectors for each location $\vec{x}$. Then,

$$\mathcal{S}_0(\vec{x}_s, \theta_i) = \begin{cases} 1 & \text{if } \exists_{v \in V(\vec{x})} \Psi(v, \Theta, \theta) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where $\Psi(\vec{v}, \Theta, \theta)$ is the "angle binning" indicator function as previously described. Using Equation 4, we obtain $\mathcal{S}$ from $\mathcal{S}_0$.

The conversion from an RFT (or filter) to an visually-intuitive vector flow representation is also straightforward.

$$V(\vec{x}) = \{\langle \cos\theta, \sin\theta \rangle | \mathcal{S}(\vec{x}, \theta_i) > \tau\} \quad (7)$$

where τ is a threshold parameter that can be adjusted to deal with noisy measurements. In our results, we chose $\tau = 0$. Note that RFT can handle multiple directions of flow at a single location, so each location maintains a set of direction vectors, V, which can be visualized, as seen in Figure 4.

So far, we have introduced the RFT and described how it can be used to query for and examine the motion patterns in data. In the next section, we discuss how we obtain the input data for the RFT for the specific problem of NKT cell motion analysis.

3. NKT Cell Detection

A rapidly growing body of evidence indicates that NKT cells are important mediators of host defense in response to bacterial and viral infection [3], as well as in defense against tumors. Recent work has demonstrated that NKT cells exhibit sentry behavior, patrolling the liver vessels. It is an open question as to how this behavior changes in the presence of signals, such as tumors in the tissue.

In Section 1, we discussed methods which estimated the flow in video without explicit object tracking. These methods would not be suitable for NKT cell flow estimation due to the erratic motion patterns of the cells. NKT cells move both up- and down-stream and rapidly change directions. So, we do not expect a spatially consistent motion pattern over the field of view and a simple parametric model would be insufficient.

In this section, we describe how we use the RFT for NKT cell motion analysis. Researchers have extensively studied the microcirculation of the liver under a variety of conditions using intravital microscopy, which is a method of visualizing the microcirculation *in vivo*. However, the resulting images (e.g., Figure 1) are challenging to analyze. The first step is to detect, as best as possible, the cells in these images.

Much of the work in blood cell detection is based on shape [9, 7]. However, NKT cells exhibit a large variation of shape. The top row of Figure 5 shows an example of a single manually-tracked cell over the course of 7 frames in a video. Given the large variation in the shape of a single cell over time and between cells, we chose to address this problem using a combined shape and appearance approach.

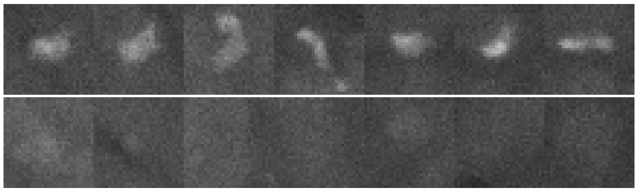


Figure 5. Samples used for training the cell detection. The top row shows positive (cells) examples and the bottom row shows negative (non-cells) examples.

We use a window-based approach to cell detection; for each pixel location, $\vec{x}$, we examine pixels in an $l_w * l_w$ window centered at $\vec{x}$. Figure 5 shows 14 small rectangular cropped regions from an *in vivo* microscopic image of a mouse liver, similar to that shown in Figure 1. The top row shows examples of regions manually classified as cells and the bottom row shows tissue (non-cell) examples. For each image window, we collect the following features (with the justification in parentheses):

- *Intensity Mean* - (NKT cells are generally brighter than the background.)
- *Temporal Difference Mean* - (NKT cells are generally moving, so the difference from the average image in a 5-frame window should be higher.) For each window, we calculate:

$$\mu_t = \frac{1}{l_w^2} \sum_{\vec{x}_j \in I_w(\vec{x})} |I_w^t(\vec{x}_j) - J_w^t(\vec{x}_j)| \quad (8)$$

$$\text{where } J_w^t(\vec{x}_j) = \min_{t'=[t-2,t+2]} I_w^{t'}(\vec{x}_j) \quad (9)$$

- *Temporal Difference Variance* - (The difference from the temporal mean image should more uniform in tissue.)
- *Radial Mean* - (NKT cells are generally brighter in a circular area around the center of the cell and darker outside.) For each image window, we calculate:

$$\mu_r = \frac{1}{l_w^2} \sum_{\vec{x}_j \in I_w(\vec{x})} \Gamma(\vec{x}, r) I_w(\vec{x}_j) \text{ where} \quad (10)$$

$$\Gamma(r_1, r_2) = 1 \text{ if } |r_1 - r_2| < \gamma, \text{ else } 0 \quad (11)$$

- *Gradient Magnitude Mean* - (Image windows with NKT cells often feature prominent edges.) So, we calculate:

$$\mu_\nabla = \frac{1}{l_w^2} \sum_{\vec{x}_j \in I_w(\vec{x})} |\nabla I_w(\vec{x}_j)| \quad (12)$$

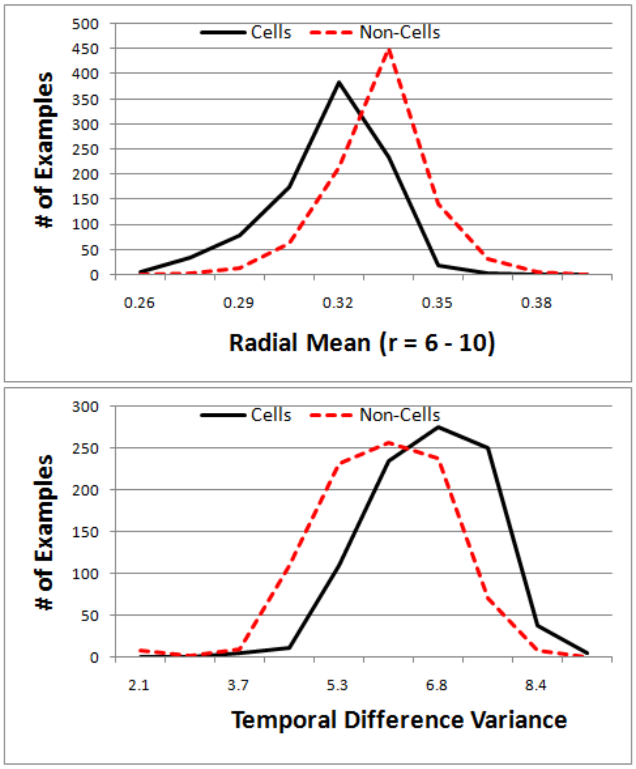


Figure 6. Distribution of training data for cell classification for two of the features we used for boosting.

For 1862 manually classified image windows, we show the distribution of positive (cell) and negative (tissue) examples for some of these features in Figure 6. For many of the features, the distribution of cell pixels does not greatly differ from that of non-cells and a linear classifier in feature space only produces slightly better than chance results. In this section, we briefly discuss how we train a discriminative classifier using boosting to combine weak classifiers on these features into a strong cell detector, in a manner similar to [6].

3.1. Boosting

Boosting is a binary classification method that combines a set of weak classifiers into a single strong classifier. Adaboost [4], the exemplar of this class of algorithms, takes as input a set of N labeled training examples $\{\langle \vec{f}_1, y_1\rangle, \ldots, \langle \vec{f}_N, y_N\rangle\}$, where $y_i \in \{1, -1\}$ is the label (e.g., cell versus non-cell) for example $\vec{f}_i$. Here, $\vec{f}_i$ is a vector of feature values for image region i, as described in the previous section. The resulting strong classifier is a weighted combination of weak learners:

$$H(\vec{f}_q) = \Sigma_{t=1}^{T} \alpha_t h_t(\vec{f}_q) \tag{13}$$

where α_t is the weight of weak classifier h_t. In our experiments, we used the GML AdaBoost Matlab Toolbox (MSU Graphics & Media Lab, Computer Vision Group, `http://graphics.cs.msu.ru`). For our weak learners, we chose classification and regression trees (CART) with at most 3 splits. For each round of boosting, at each level of the decision tree, we find the feature and value that provides the (weighted) optimal linear decision boundary. Using 1862 manually labeled examples, 931 were selected as the training set and the remainder used for validation. Training the learner took 5 iterations of boosting. The output classifier performed at 87% accuracy on the validation examples. Figure 7 shows classification results on a region of liver from our data set.

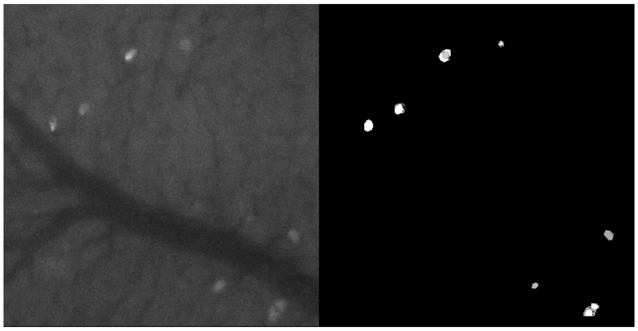


Figure 7. Classification results for a frame containing NKT cells in a mouse liver. The intensity of the output in (b) represents the confidence of the classification.

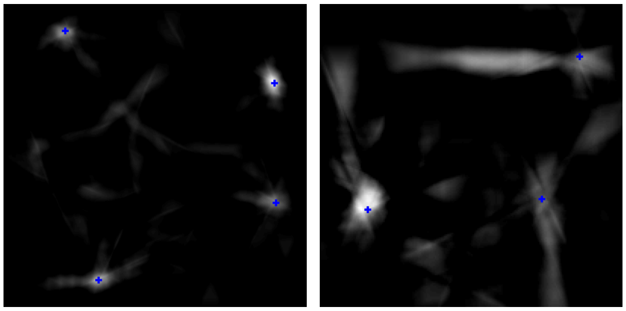
4. Results

In this section, we demonstrate results of the Radial Flow Transform on simulated data and show some preliminary efforts of using the motion analysis application on biomedical data.

4.1. Simulated Data

In order to test the ability of this method to find specific cell motion patterns, we generated synthetic image sequences by randomly placing n_c cells and n_t interest points in the image field. The interest points were static and each interest point induced a specified behavior in all nearby cells. Each synthetic cell is assigned a general direction, $\vec{v}$, which is corrupted by Gaussian noise at each time step. To simulate some of the hypothesized motion patterns of cells around tumors, we defined the following interest points:

- *Attracting.* These attract any cells within a certain radius. (For our tests, this value was roughly 10% of the width of the image.)
- *Sentry.* These sequentially attract then repel cells to induce a “sentry-like” motion of cells within a certain radius.



(a) *Attracting* Simulation Results (b) *Sentry* Simulation Results

Figure 8. Pattern match scores for two simulated data sets. In each plot, the intensity of the image represents the pattern match score and the locations of the interest points are marked with crosses. In the left figure, the interest points induced the *attracting* pattern and in the right figure, the *sentry* pattern. For each experiment, we used the corresponding filter.

Figure 8 shows results for two experiments using the simulated data. The results in Figure 8(a) are from an experiment with a 300*300 video with 50 frames, 20 cells, and 4 *attracting* interest points with influence radius 50. The results in (b) are from a similar experiment with 3 *sentry* interest points. The intensity of image represents the pattern match score. For the experiments we use the corresponding RFT filters depicted in Figure 4.

Table 1 summarizes the results for additional experiments using simulated data. Each simulation was performed on a 500*500 field with 40 frames of motion. The support of each RFT filter was 61*61. For each experiment, the percentage of interest points correctly detected is reported for each threshold, T_p. We consider an interest point correctly detected if its pattern matching score falls above the T_p percentile. The RFT filter used in these experiments corresponds to the type of the interest point.

			Pattern Match Threshold, T_p			
Type	Cells	IP	99 %	95%	90%	80%
A	10	4	50%	100%	100%	100%
A	20	6	66%	83%	100%	100%
A	25	6	83%	100%	100%	100%
S	10	4	25%	50%	75%	100%
S	20	6	17%	66%	66%	100%
S	25	6	17%	33%	83%	83%

Table 1. Summary of experiments using simulated data for attracting (A) and sentry (S) interest points (IP). For each experiment, the percentage of interest points correctly detected is reported for each threshold, T_p. An interest point is considered correctly detected if its pattern matching score falls above the T_p percentile. The RFT filter used in these experiments corresponds to the type of the interest point.

4.2. Medical Data Visualization

The RFT can be used by microcirculation researchers to quickly understand the flow patterns at a given location. In these experiments, we highlight the visually-intuitive output of RFT by comparing the vector flow field output to traditional optical flow for two motion analysis problems in microcirculation research. We tested these methods on red blood cells (RBC), which generally move in the direction implied by the vessels and NKT cells which move more erratically. Figure 9 shows visualization results for both red blood cell and NKT cell motion. For each cell type, we show a cropped view of the entire field, an overlay of the motion of the cells in the video, optical flow results, and the RFT visualization. To calculate optical flow, we used the Lucas-Kanade method between sequential frames. We then reported the mean optical flow, only counting pixels locations if the optical flow magnitude was greater than 0. These results demonstrate the power of the RFT to describe multiple dominant directions of motion in a visually intuitive way. Also, in addition to visualization, these results could be used to initialize vessel segmentation.

4.3. NKT Motion Analysis

Using an Olympus IX70 fluorescence inverted microscope, the livers of cxcr6$^{gfp/+}$ mice were imaged using a specially constructed stage. The NKT cells are made visible using a green fluorescent protein (GFP) filter. The injected tumor cells appear using a red filter. The liver surface is imaged every minute for one hour at 10x magnification. For each capture cycle, the filter is altered to image both NKT and tumor cells.

There is ongoing research in the microcirculation community regarding the motion patterns of NKT cells in the presence of tumors. Here we demonstrate the results of querying using the real NKT data. For this example, we created a 101*101 RFT filter for the sentry pattern and applied it to this video. Figure 10 shows the pattern matching scores with the tumor location marked with a blue cross. The score for the region around the actual location of the tumor was in the 99th percentile of our pattern match score results.

5. Conclusions and Future Work

Learning the motion patterns of cells *in vivo* is a problem challenging microcirculation researchers. Here, we presented the Radial Flow Transform which can be used to query for and describe motion patterns in microscopy imagery. We demonstrated results on simulated data and showed some results on NKT data which show promise for future use in research on cell motion analysis. In the future, we hope to use the RFT to seed vessel segmentation algorithms and extend the matching operation to allow for

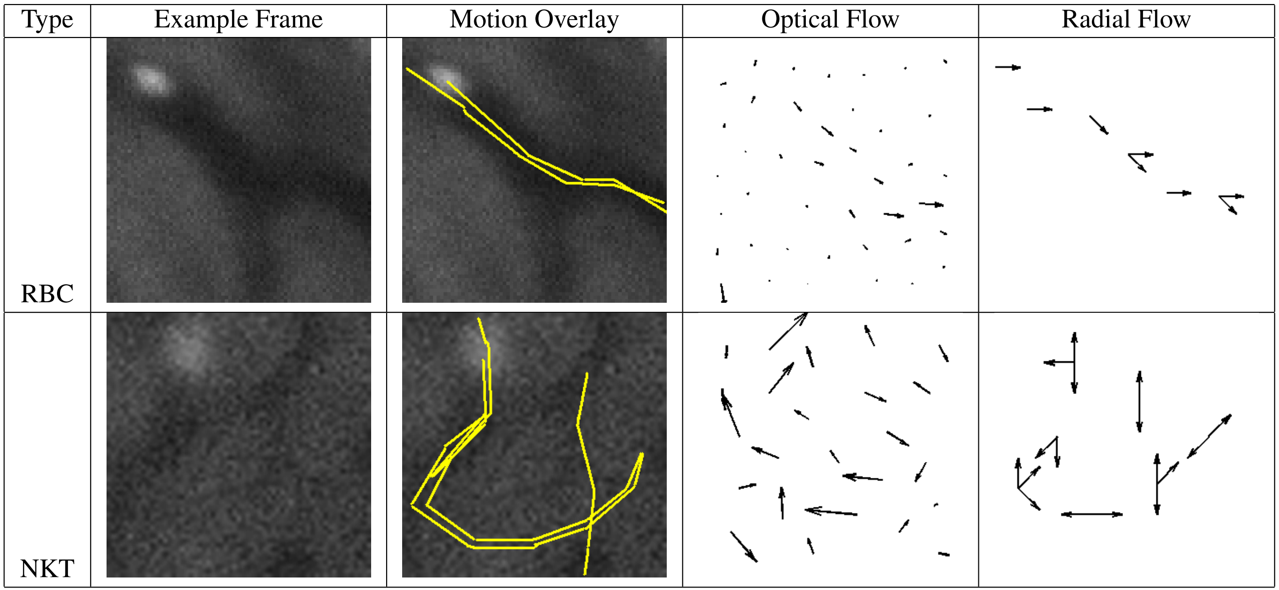
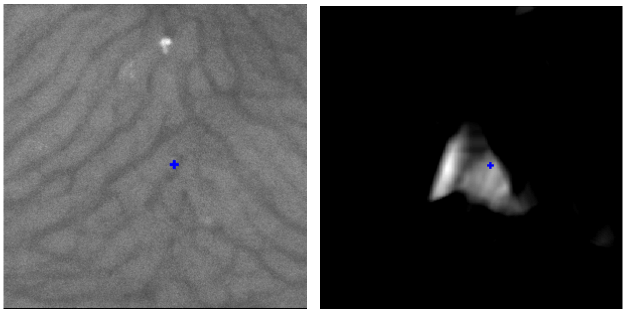


Figure 9. Comparison of the radial flow with optical flow for medical image analysis. For both red blood cells and NKT cells, this table shows the visualization of the output.



(a) Example Frame

(b) Pattern Match Score

Figure 10. Results of searching a sequence on real NKT data for the sentry pattern using the Radial Flow Transform. The score of the location of the tumor (marked with the cross) was in the 99^{th} percentile of our results.

rotationally-invariant pattern matches. We also wish to provide an application to allow users to easily describe and query for a desired motion pattern.

References

[1] O. Debeir, P. Van Ham, R. Kiss, and C. Decaestecker. Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. *IEEE Trans. on Medical Imaging*, 24(6):697–711, June 2005.

[2] E. Eden, D. Waisman, M. Rudzsky, H. Bitterman, V. Brod, and E. Rivlin. An automated method for analysis of flow characteristics of circulating particles from in vivo video microscopy. *IEEE Trans. Med. Imaging*, 12(8):1011–1024, 2005.

[3] M. Exley and M. Koziel. To be or not to be NKT: Natural killer T cells in the liver. *Hepatology*, 40:1033–1040, 2004.

[4] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT '95: Proc. of the 2nd European Conf. on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag.

[5] S. Hinz and A. Baumgartner. Road Extraction in Urban Areas Supported by Context Objects. In *Int'l Archives of Photogrammetry and Remote Sensing*, volume 33, pages 405–412, 2000.

[6] S. Mallick, Y. Zhu, and D. Kriegman. Detecting particles in cryo-EM micrographs using learned features. *Journal of Structural Biology*, 145(1-2):52–62, Jan-Feb 2004.

[7] D. Mukherjee, N. Ray, and S. Acton. Level set analysis for leukocyte detection and tracking. *IEEE Trans. on Image Processing*, 13(4):562–572, April 2004.

[8] R. Pless and D. Jurgens. Road extraction from motion cues in aerial video. In *Proc. of the ACM Conf. on Geographic Information Systems*, pages 31–38, 2004.

[9] N. Ray, S. Acton, and K. Ley. Tracking leukocytes in vivo with shape and size constrained active contours. *Medical Imaging, IEEE Transactions on*, 21(10):1222–1235, Oct 2002.

[10] M. Roth and W. T. Freeman. Orientation histograms for hand gesture recognition. In *Intl. Workshop on Automatic Face and Gesture Recognition*, 1995.

[11] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.