Automated Kymograph Analysis for Profiling Axonal Transport of Secretory Granules

Amit Mukherjee[a], Brian Jenkins[b], Cheng Fang[b], Richard J. Radke[a], Gary Banker[b], Badrinath Roysam[a,*]

[a]*Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy NY 12180*

[b]*Jungers Center of Neuroscience Research, Oregon Health and Science University Foundation, Portland, OR 97239*

Abstract

This paper describes an automated method to profile the velocity patterns of small organelles (BDNF granules) being transported along a selected section of axon of a cultured neuron imaged by time-lapse fluorescence microscopy. Instead of directly detecting the granules as in conventional tracking, the proposed method starts by generating a two-dimensional spatiotemporal map (kymograph) of the granule traffic along an axon segment. Temporal sharpening during the kymograph creation helps to highlight granule movements while suppressing clutter due to stationary granules. A voting algorithm defined over orientation distribution functions is used to refine the locations and velocities of the granules. The refined kymograph is analyzed using an algorithm inspired from the minimum set cover framework to generate multiple motion trajectories of granule transport paths. The proposed method is computationally efficient, robust to significant levels of noise and clutter, and can be used to capture and quantify trends in transport patterns quickly and accurately. When evaluated on a collection of image sequences, the proposed method was found to detect granule movement events with 94% recall rate and 82% precision compared to a time-consuming manual analysis. Further, we present a study to evaluate the efficacy of velocity profiling by analyzing the impact of oxidative stress on granule transport in which the fully automated analysis correctly reproduced the biological conclusion generated by manual analysis.

[*]Corresponding author: roysam@ecse.rpi.edu

Keywords:
Particle Tracking, Brain Derived Neurotrophic Factor, Kymograph, Tensor voting

1. Introduction

Quantitative analysis of transport of organelles in neurons is important to several biomedical investigations. For instance, the transport of membranous organelles along axons is of vital interest in studies of neurodegenerative diseases, especially Huntington's and Alzheimer's [1–4]. For example, Ferrer et al. in [3] found that a key role of huntingtin (HTT) protein is to regulate transport of secretory granules containing Brain-derived Neurotrophic Factor (BDNF), and that loss of this function may contribute to pathogenesis in Huntington's disease. These processes can now be imaged by time-lapse microscopy of live cultured neurons using a combination of fluorescent protein labeling and sensitive instrumentation [5]. However, tracking of granules in the resulting video sequences is a laborious, time-consuming, and subjective task that is currently performed manually. In this paper, we propose robust, automated computational tools to extract measurements of granule transport accurately, objectively, and rapidly. Such automated analysis of these time-lapse movies can enable high-throughput applications such as screening of interventional drug candidates.

Automated tracking of fluorescently tagged BDNF granules from time-lapse video image sequences is challenging for several reasons. First, the granules are tiny ($0.2 - 0.5\mu m$), occupying only a few pixels in images when the magnification is chosen to capture at least $150\mu m$ of the axon length in a single video frame. They move very rapidly compared to currently practical video sampling rates (about 2 Hertz). These rates represent a trade-off between several factors including the need for granules to be distinguishable from the background noise and for capturing their dynamic activity. Even then, the granules appear as barely-discernible changes in image intensity, lacking temporal persistence, and are especially difficult to discern when crossing other granules. Sometimes, the axon segments dip in and out of the depth of field of the microscope causing the granules to "disappear" temporarily. The granules exhibit a high variability in morphology and appearance due to motion blur induced in the images as well as the point spread function of the microscope. In addition, some granules can move much faster than others (e.g., in some cases a distance several times their diameter is covered within two successive frames). Sometimes, they pause for brief periods before resuming movement. The granules travel in both directions (anterograde/retrograde) along the axon, and their paths often cross. Figure 1(a-c) shows cropped (100-by-100 pixel) sections from frames of a video where moving granules are indicated by arrows. Figure 1(a) shows two granules traveling in opposite directions and crossing each other, Figure 1(b) shows a change in apparent granule morphology while traveling due to imaging artifacts, and Figure 1(c) shows two granules passing bright stationary granules during their traversal.

The granule movements show certain regularities in their movements that provide a valuable clue for locating and tracking them. Their movements are not altogether erratic — they exhibit some consistency in velocity. This is due to the fact that these movements are caused by attached molecular motors, whose speed is determined by the geometry of the vesicle, and molecular properties of the motor proteins, which are relatively stable over time.

A *kymograph* is a distillation of a video sequence into a spatio-temporal map that summarizes granule motion along a one-dimensional path defined by the central axis of an axon segment. For example, the kymograph in Figure 1(e) is created from an axonal segment shown in Figure 1(d). The kymograph has time along the x-axis and the length along the axonal path along the y-axis. The intensity of each spatio-temporal pixel in the kymograph represents the likelihood of an organelle being present at that location and time. Thus, the kymograph mapping provides a reduced-dimensional spatio-temporal representation that isolates only the relevant portion of the video. For the purpose of visualization, we render a kymograph using inverted intensities, i.e., bright background and dark foreground. For the purpose of illustration, Figure 1(e) shows the representative granules in Figure 1(a-c) overlaid on the kymograph, showing that they follow near-linear trajectories.

The kymograph representation offers multiple advantages. First, this formulation avoids the need to detect individual granules in individual video frames. Instead, it provides a quantitave approach for estimating the presence and movement of granules based on multiple video frames. This is particularly useful in the case of BDNF granules where the motion can only be ascertained after observing a granule over a sufficient number of frames. A second advantage of the kymograph is its ability to express a path constraint for the particle trajectories, enabling far more reliable automated analysis than would be possible without such a constraint. A helpful side effect is to eliminate irrelevant pixels without loss of relevant information. It focuses the analysis on the interesting longitudinal component of motion of granules rather than the transverse motion that is both negligible and uninteresting. The central axis also provides a reference path along which the relative velocities of the granules can be compared and analyzed. As discussed in Section 2, kymographs are a common tool in transport analysis, but few automated algorithms exist for their analysis.

This paper presents an automated system to: (i) generate kymographs from a selected section of axon of a cultured neuron in multi-channel image data, (ii) enhance the kymographs for subsequent image analysis, and (iii) process the kymographs to quantify the aggregate velocity patterns of fluorescently tagged secretory granules in axons. Section 3 details the basic construction of kymographs from time-lapse image sequences and the initial processing steps. Section 4 describes a new method to enhance the kymograph, improving the visual continuity of trajectories using a method based on orientation voting. Section 5 describes an algorithm for processing the enhanced kymograph to automatically extract the set of granule trajectories. In Section 6, we validate these trajectories against a manual observer, and show how automatically extracted measurements of trajectory length, trajectory counts, velocity distribution, and anterograde-to-retrograde bias can be used to profile granule motion patterns. We present a study of the impact of oxidative stress on granule transport, in which the fully automated analysis correctly reproduces biological conclusions generated by tedious manual analysis. Section 7 concludes the paper with discussion of potential applications of our method. The steps of the algorithm are pictorially shown in Figure 2.

2. Related Work

Traditional video tracking and motion analysis algorithms (e.g., the mean shift algorithm [6]) are unsuitable for the BDNF granule tracking problem because of some of the peculiarities described in Section 1. Formulating the problem as frame-to-frame tracking of blobs would require accurate detection of the granules at each frame, which can be unreliable for BDNF granules due to several reasons: their small size, low signal-to-noise ratio, high speeds at lower frame rates, and frequent crossing of particle trajectories. Jaqaman et al. [7], described single particle tracking methods for time-lapse sequences that can handle erratic Brownian motion. However, these methods would probably be ineffective for BDNF granules where the average displacement of granules between frames is greater than average nearest-neighbor distance between granules in the same frame. Statistical tracking methods [8] and methods using particle filters [9] also would require detection of objects prior to finding their trajectories. Another approach for tracking uses a spatiotemporal image [10, 11] that detects a temporal trace of particles in a composite (x, y, t) map. This approach is advantageous because contextual information from multiple frames is directly available to the analysis at a given frame. For example, Padfield et al.[11] showed that using spatiotemporal images can improve segmentation and localization of multiple cells, for tracking cell cycle phases and enabling movements to be estimated accurately. Kymographs can be considered as a special case of this spatiotemporal method for motion analysis and detection. The idea of using kymographs of video sequences to form a spatiotemporal image representation has been proposed in several prior works [12–14], although they are constructed differently than in this paper. Welzel et al. [12] evaluated axonal transport by cross- and auto-correlation of kymograph columns. Ludington et al.[13] generated kymographs by hand and analyzed them using the Radon transform to detect peak velocities of particles. Kymographs have also been used to track image features over time; for example, the algorithm of Smal et al.[14] used a variable-rate particle filter to enhance the accuracy of the extracted edges corresponding to the tip of micro-tubules from a kymograph-like image representation. Our framework for kymograph analysis is illustrated in Figure 2 and discussed in detail in the next three sections. We begin with the basic process of generating the kymograph for BDNF granules, followed by a temporal sharpening filter to highlight fast-moving granules.

3. Constructing the Kymograph

The transport of BDNF granules is confined within an axon, which forms a narrow tube that can be labeled with an appropriate fluorescent marker (e.g., enhanced green fluorescent protein, eGFP) and detected in a separate channel (Figure 3a). For biological analysis, the movement of the granules in a direction parallel to the length of the axon is important, but the motion component that is normal to the micro-tube is uninteresting and negligible. This fact can be used to our advantage by collapsing the frames in the video sequence along a one-dimensional curved axis, so that granule movements along that axis can be studied and analyzed in a lower-dimensional space.

During the imaging process, the orientation of the axon of interest relative to the nucleus is not always obvious from the limited view of axon segment that captured by the camera. For this reason, we ask the investigator to indicate a pair of points on the axon segment - one point is closer to the neuronal soma and is termed the "proximal point" and the other point is termed the "distal point". The granule transport patterns along the specified segment of the neuronal anatomy delimited by the proximal and distal points are then examined.

The first step in the construction of a kymograph is to trace the central axis of the intervening axon segment accurately and automatically. For estimating the central axis, we follow a method similar to that described in [15] and [16]. Denoting the eGFP image of axon segment as $J(x)$, where x refers to the spatial coordinates, a vesselness potential map [17] is first computed. The vesselness potential map uses a kernel function to detect longitudinal structures at each pixel of the image using the eigenvalues of the local Hessian estimated at scale σ. The scale of the Hessian is selected so that the response of the vesselness potential is maximized. The central axis is then estimated by minimizing the integral of the vesselness potential over a set of simple, non-intersecting curves connecting the proximal and distal points on this potential map. That is, if $\xi(x)$ is the vesselness potential, then the central axis C is chosen to minimize the function $E(C) = \int\limits_{s \in C} \xi(s)ds$. The Eikonal equation with boundary conditions (from the proximal and distal points) is solved using the Fast Marching algorithm [18, 19] for the minimization. An example central axis is shown in Figure 3(a) as a blue line.

Next, the basic kymograph is generated automatically using the central axis obtained in the previous step and the video sequences of granule transport denoted by $I(x, t)$. As shown in Figure 4(a), the central axis C is parameterized into discrete steps denoted by s_i along the axon length. The kymograph, denoted $K(t, s_i)$, encodes the likelihood that a granule is present at discrete position s_i and the video frame with time index t. For each time frame, we dilate the axon centerline by a user defined width τ (typically 4-8 pixels). We then compute compartmental regions $X(s_i)$ that form strips along the central axis by selecting all the points in the dilated region that are closer to s_i than any other $s_{j \neq i}$ (see Figure 4). Following this step, we select the pixel location $x^*_{s_i,t}$ from each set $X(s_i)$ that has the maximum pixel intensity, i.e., $x^*_{s_i,t} = \arg\max_{x \in X(s_i)} I(x, t)$.

Finally, we apply a temporal filter on pixel intensities to construct the final kymograph using the following formula:

$$K(t, s_i) = \frac{1}{2}(I(x_{s_i,t-1}) - 2I(x_{s_i,t}) + I(x_{s_i,t+1})). \tag{1}$$

The temporal sharpening filter accentuates the moving granules, while filtering out the stationary granules that are generally brighter than the moving ones. The effect of the temporal sharpening filter is illustrated in Figures 4(b) and (c). The stationary particles (horizontal lines) are diminished and fast-moving particles (diagonal lines) are accentuated.

The motion of the granules as seen on the kymograph can be interpreted as follows. If the granule is moving from the proximal point to the distal point, it is said to be engaged in anterograde motion, and the corresponding trajectory appears with a positive slope. If the granule is moving from the distal point to the proximal point (i.e., retrograde motion), then the line appears with a negative slope. Trajectories of faster moving granules have larger absolute slopes. Our next goal is to enhance the kymograph to make it easier to extract the motion trajectories.

4. Refining the Kymograph Using Orientation Voting

The granule trajectories that are revealed in the kymograph as intensity ridges have a systematic velocity pattern due to the characteristics of the molecular motors (kinesins and dyneins) that drive them. This regularity in the velocity patterns of the granules motivated us to develop a voting scheme based on the orientation of neighborhood points to refine the ridge-like structures in the kymograph. Detecting such structures, is challenging due to the presence of gaps, noise and ridge intersections. The original video is typically captured at a slow frame rate to achieve sufficient exposure times. As a result, the fast moving granules have little or no overlap between successive frames. This creates gaps along the ridges (see Figures 3b, 4b), making it difficult for conventional ridge detection algorithms that rely on eigen-analysis [20, 21] or matched filters [22].

To overcome this difficulty we propose a novel voting based algorithm that refines the kymograph, making it easier to automatically extract ridges. Similar to voting based methods [23–25], the proposed algorithm uses an inferencing method inspired by Gestalt principles of proximity and good continuation. The first step is to compute what we term the orientation distribution function (ODF) at each pixel of the kymograph, which is simply a set of discrete line integrals over orientations as shown in Figure 5, weighted by a bell-shaped kernel given by:

$$\Omega(u, \theta) = \int_{p=-l}^{p=l} K(u + r_\theta(p)) e^{\frac{-p^2}{2l^2}} dp, \tag{2}$$

where $u = (t, s_i)$ is a point in the kymograph, θ is a discrete orientation, $r_\theta(p) = [p\cos(\theta), p\sin(\theta)]$ and l is the width of the ODF. Any ridge passing through u creates a peak in Ω at the angle θ coinciding with the orientation of the ridge. Note that the line integration is not done with θ perpendicular to the time axis because it would imply a direction of infinite velocity, which is not possible. A schematic of the kernel is shown in Figure 5(a). Importantly, the ODF can represent multiple peaks that occur at intersections of multiple ridges (i.e., overlapping granules) which would not be possible using an eigen-analysis approach that assumes only one orientation per pixel.

The ODFs are used in a voting scheme to fill in the gaps and inconsistencies in the appearance of the motion paths, while preserving ridgelines at the trajectory intersections. The basic idea is that each point in the neighborhood of a given point u casts a "vote" for what the refined intensity at u should be. The votes are constructed so that points on the ridges are amplified, points between ridges are down-weighted, and gaps in ridges are filled from strong ridges on either side.

Similar to concepts from tensor voting [26], the magnitude of the vote is computed by comparing the ODF $\Omega(u, \theta)$ with a uniform distribution over θ that represents an isotropic ODF. In particular, we define a quantity $\Psi(u, \theta)$ obtained by removing the isotropic part of the distribution from $\Omega(u, \theta)$, that is,

$$\Psi(u, \theta) = \Omega(u, \theta) - \frac{1}{q}, \tag{3}$$

where q is the number of bins used to discretize the ODF. The value of $\Psi(u, \theta)$ is positive for peaks in the ODF that indicate ridge directions and is negative for valleys between peaks. During the voting process, this results in a positive vote cast by a neighboring pixel if there is agreement in the ridge orientations between the voter and the receiver, and a negative vote would be cast if there is an orientation mismatch. We also define another quantity $\Psi_+(u, \theta)$ that is equal to $\Psi(u, \theta)$ if $\Psi(u, \theta) > 0$ and 0 otherwise. Intuitively, $\Psi_+(u, \theta)$ consists of only the positive anisotropy that indicates the presence of a ridge, but does not penalize its absence.

The voting also depends on a neighborhood decay function, denoted $N_{u,v}(\theta)$ that defines the spatial relationship between the voter v and the receiver u, so that the neighborhood points that are further away from or angularly misaligned with u have less influence on the vote. Here, we use the form $N_{u,v}(\theta) = \|d_{u,v}\|^m G_\sigma(\angle d_{u,v} - \theta)$, where $d_{u,v} = u - v$ and G_σ is a Gaussian with standard deviation σ applied to the angular discrepancy between θ and the direction of vector $d_{u,v}$. The neighborhood parameters m and σ are set by the user based on the desired angular resolution (i.e., velocity resolution for the granules) and permissible sizes of gaps or inconsistencies in intensities. For the results in this paper, we used $m = -0.1$ and $\sigma = 15$ degrees. Figure 5(b) shows neighborhood decay functions for a few sample orientations.

Figure 5(c) presents a schematic of the voting procedure. Here, u denotes a candidate spatiotemporal point in the kymograph, and v and w denote two points in its neighborhood. Votes cast from point v for each discrete orientation are accumulated at point u . The magnitude of the vote depends on three quantities: the modified ODFs $\Psi_+(u, \theta)$, $\Psi(v, \theta)$ and the neighborhood decay function $N_{u,v}(\theta)$. With this notation, the vote cast by v on u is denoted $V(u, v)$, and computed by:

$$V(u, v) = \int_\theta \Psi_+(u, \theta)\Psi(v, \theta)N_{u,v}(\theta)d\theta, \tag{4}$$

Note that we have used Ψ_+ for u and Ψ for v, which means that the receiver point u seeks votes only for directions for which positive anisotropy is found (i.e., ridges are present), whereas the voters v cast both positive and negative votes to the receiver u.

Considering the examples in Figure 5(c), $\Psi_+(u, \theta)$ and $\Psi(v, \theta)$ have positive anisotropies in the retrograde direction and the corresponding neighborhood decay function $N_{u,v}(\theta)$ also shows a peak near the angle θ that matches with the positive anisotropies of $\Psi_+(u, \theta)$ and $\Psi(v, \theta)$. Thus a positive vote would be cast on u from v . On the other hand, while $\Psi(w, \theta)$ has a positive anisotropy in the retrograde direction similar to u , the peak of the neighborhood function $N_{u,w}(\theta)$ does not match the anisotropy direction, making the weights almost negligible at those angles. Therefore, a near zero vote is cast from w to u for an angle θ in this case. However, if a neighboring point has a negative anisotropy for any θ and the neighborhood decay function is nonzero for that angle, a negative vote is cast on u . The total votes from all neighbors are accumulated over the entire neighborhood as follows:

$$\widehat{K}(u) = \sum_{v \in Nb(u)} V(u, v), \tag{5}$$

where $\widehat{K}(u)$ denotes the kymograph $K(u)$ after the voting process.

In this way, θ 's for all positive anisotropies of u accumulate positive as well as negative votes from the neighboring points that have non-negligible weights in the decay function according to (4). In the end, the receiver point u appears to be a ridge if the sum of positive votes exceeds that of the negative votes. In other words, if the neighborhood of u has a sufficient number of matching peaks in its anisotropy direction, it receives a large number of positive votes even if it was not originally a ridge point due to a gap. On the other hand, if the receiver point u is a noisy pixel or small "island" that does not belong to any trajectory in the kymograph, it receives a large number of negative votes from its neighbors and gets suppressed.

As a final step of refinement, we use the Weingarten matrix [27] to thin the ridgelines and sharpen the $\widehat{K}(u)$ image. In our case, the ridgelines are obtained using a voting process and their cross-section shows a unique peak with monotonic descent on either sides. The Weingarten representation extracts the peak of the ridgeline using available curvature information instead of the height of the ridge, and therefore helps recover the low-contrast ridgelines. In contrast to Hessian-based ridge operators that use the second order derivative matrix, the Weingarten operator uses first and second derivatives, and its response is less sensitive to the height (contrast) and cross-sectional width (scale) of the ridge-lines. The Weingarten matrix W, is expressed as

$$\begin{aligned} W(u) &= F_2 F_1^{-1}, \\ F_1 &= \begin{bmatrix} 1 + \widehat{K}_t^2 & \widehat{K}_t \widehat{K}_s \\ \widehat{K}_t \widehat{K}_s & 1 + \widehat{K}_s^2 \end{bmatrix}, \\ F_2 &= \frac{-1}{\sqrt{1 + \widehat{K}_t^2 + \widehat{K}_s^2}} \begin{bmatrix} \widehat{K}_{tt} & \widehat{K}_{ts} \\ \widehat{K}_{ts} & \widehat{K}_{ss} \end{bmatrix} \end{aligned} \tag{6}$$

where F_1 and F_2 denote the first and second fundamental forms and the subscripts denote derivatives with respect to the spatial (s) and temporal (t) axes, respectively. The largest magnitude eigenvalue of $W(u)$ at each pixel gives us the refined kymograph $K_{refined}$ that we use for detecting trajectories as explained in the next section. The pixel values of this refined kymograph can be interpreted as saliencies, or likelihoods that a trajectory is present at a given point.

Figure 6 illustrates the overall voting scheme for a small section of a sample kymograph. A, B and C are 3 points on the kymograph and its refinement. They are chosen so that A lies on a valley between two trajectories, B lies on a trajectory and C lies at the junction of two trajectories. Figure 6(a) shows these locations on the raw kymograph, whereas Figure 6(b) and Figure 6(c) show these locations on $\widehat{K}$ and $K_{refined}$, respectively. The ODFs before voting are shown in Figure 6 (d)-(f), where the isotropic ODFs are shown using a circle. The values of θ's where the radius is greater than the isotropic value constitute positive anisotropy and those with lesser radius represent negative anisotropy. The result of voting is shown in Figure 6(g)-(i) which is equivalent to the vector of values for each θ before the integration step in (4).

The accumulation of votes generates a resultant vote that has a high value only when there is a spatial and angular alignment of the underlying ridgeline(s). This ensures sharp and narrow peaks at the center of the ridgelines that robustly indicate the location and direction of the ridges. For example, in Figure 6(d), the ODF at Point A has some local anisotropy that is due to the nearby ridges. However, Point A does not lie on a ridge-line and therefore the peak was suppressed after the voting process because the orientation of Point A did not match with other ODFs in the direction of the peak. Point B, on the other hand, collected enough votes from the neighbors that lie in the direction of its ODF peak and it survived the voting process. Similarly, Point C collected votes from two ridges passing through it and retained both ridgelines. Since the direction of the ridges in the kymograph is an indicator of the speed of granules in the video, we can also estimate granule velocity using this technique. The next section describes methods to extract the ridgelines as curves, thus recovering the motion trajectories of the granules.

5. Multiple Trajectory Detection

Extracting motion trajectories of granules amounts to extracting curves along the ridgelines of $K_{refined}$. Our approach to multiple trajectory detection is organized into two steps; the first step incrementally grows multiple trajectories along ridgeline pixels and the second step joins fragmented trajectories using multiple regression to recover more complete motion trajectories.

Felzenszwalb et al.[28] proposed a framework for detecting multiple curves in an image based on principles of minimum set covering, a simple yet powerful approach. Their greedy approximation algorithm ensured that the total cost of curve-coverage per unit area of the image remains minimal. Our proposed algorithm recovers multiple granule trajectories that appear as spatio-temporal curves on the refined kymograph based on similar principles. The minimum set cover approach uses a best-first search based on a cost function of our curve model (given below) to find multiple curves that cover granule trajectories in $K_{refined}$.

We define a curve T_i as a sequence of vertices denoted by $\{u_1^i, u_2^i, ...u_{p_i}^i\}$, where p_i denotes the number of vertices in the curve. We call the tips of each curve on either end the "head" and the "tail", respectively, such that the head always points towards the positive time axis and the tail points towards the negative time axis. Our general approach is to evaluate a curve cost at each current tip, and to extend the curve at the tip with the lowest cost, until the lowest-cost tip is above a stopping threshold. To initiate the process, we compute a set of seed points, defined as local peaks in $K_{refined}$ above a given value with respect to 5x5 pixel neighborhoods. Each seed point acts as the head and tail of an initial curve. At each step, a curve is selected to grow at one of its tips, by selecting the tip with overall minimal cost computed using:

$$cost(T_i^h) \propto \frac{1}{p_i'} \left(C - \sum_{j \in \text{last } l} \log \phi_1(u_j^i, u_{j+1}^i) - \sum_{j \in \text{last } l} \log \phi_2^h(u_j^i, u_{j+1}^i) - \sum_{j \in \text{last } l} \log p_T(u_j^i) \right) \tag{7}$$

where $h \in \{head, tail\}$ and $p_i' = \min\{p_i, l\}$. Note that all the sums are taken over the last l vertices, counting from the tip in the backward (for heads) or forward (for tails) directions. If l segments are not available for short curves, all curve vertices are used in the sum. We used $l = 10$ for the experiments in this paper. In (7), C is a constant that generally makes shorter curves more costly than longer curves. The ϕ_1 term models a smoothness constraint that limits the bending of the curve between adjacent segments to 45 degrees from a predicted direction. This direction d_{pred} is computed by averaging the last l directions between consecutive segments starting from the tip under consideration. The ϕ_2^h term models a feasibility constraint, which prohibits the curve from becoming normal to the time axis (this would mean that a granule is located at two positions at the same time). The p_T term is a likelihood function for the intensities of pixels on trajectories, computed using $K_{refined}$.

In our implementation, we approximate ϕ_1 and ϕ_2^h in (7) using the forms shown in Figure 7(a,c,d). The smoothness term ϕ_1, defined as a function of the difference between the predicted (d_{pred}) and actual ($d = u_{j+1} - u_j$) directions, is assumed to be constant for values less than 45 degrees and 0 otherwise. Similarly, ϕ_2^{head} and ϕ_2^{tail} are assumed to be constants for directions moving strictly forward and backward in time, respectively, and 0 otherwise. In other words, the costs due to these terms remain constant as long as the smoothness and feasibility constraints are satisfied, and they become prohibitively large upon their violation, like a binary switch. The last term in eqn(7) associates the cost of the curve with the local value of $K_{refined}$, and dominates the overall cost. Since $K_{refined}$ is the largest magnitude eigenvalue of the Weingarten matrix, it can be considered being proportional to the likelihood that a pixel lies on the trajectory [27]. Therefore, we assume $p_T \propto K_{refined}$, as illustrated in Figure 7(b). This formulation ensures that the curves covering many dark pixels ($K_{refined}$ low) have a high cost and vice-versa.

The order of incremental growth of the multiple curves in the kymograph is maintained using a priority queue, as summarized in Algorithm 1. At each step, the priority queue selects the best curve tip (head or tail) based on (7) to extend the curve by a single pixel in the 8-neighborhood of the chosen tip. The process stops when the cost of the chosen tip exceeds a stopping threshold. If two curves intersect, they are merged if the angle between the "colliding" head and tail is less than 45 degrees. If the angle is greater, or if the tip of one curve intersects another curve in the middle, then the growing tip is allowed to jump over the other curve. This is accomplished by extending the search radius for the next point to the set of pixels just beyond the other curve, as described in Algorithm 1.

The final step of multiple trajectory detection is joining fragmented curves to form a more complete granule trajectory. Due to occasional high density of trajectories, there could be multiple candidates for joining to a given head (or tail). In order to select the best possible candidates for joining, we propose a multiple regression based scheme. As illustrated in Figure 8, for each curve-head, we use a search window that is directed towards the positive time

```
Input: 1) Refined Kymograph,
2) Background value $b$ (default median $K_{refined}$ )
Output: List of Trajectories
Initialize Label image $L \longleftarrow \emptyset$
Find the local peaks greater than $b$ and assign as seed points
foreach seed point do
    Assign head (forward in time) and tail (negative in time) directions
    Push each head/tail to min-priority-queue using $w$ as sorting key
    Update $L$
end
while size(min-priority-queue) $\geq 0$ do
    Pop tip $T_j^h(u)$ having least average $w$ ;      // Tip $h \in \{head, tail\}$
    if cost($T_j^h(u)$) $> -\log(b)$ then // Check termination
        continue
    radius = 1
    $v = \underset{q \in N_{radius}(u)}{\operatorname{argmax}} \big(C - \log \phi_1(u,q) - \log \phi_2^h(u,q) - \log p_T(u_j^i)\big)$
    if $L(v) == \emptyset$ then
        $L(v) \longleftarrow L(u)$
        Update head (or tail) direction and average $w$ of curve,
        Push tip $T_j^h(v)$ to the min-priority-queue
    else if v and u have matching trajectories* then
        $L(v) \longleftarrow L(u)$ , Remove $T_j^h(v)$ from min-priority-queue
        Assign all pixels with $L(v)$ to $L(u)$,
        Update head (and tail) directions and average $w$ of new $T_j^h$
    else if radius $< 3$ then
        radius = radius + 1 and repeat
end
*matching implies that one tip is head and other tail from different trajectories.
```

Algorithm 1: Multiple Trajectory Detection

axis, and covers later time points in that direction. Similarly, for each curve-tail the search window is directed towards the negative time axis, and covers earlier time points. All curves that lie inside the search window corresponding to a given head (or tail), form a candidate connection for that head (or trajectory-tail).

We denote by C_{ij} a feasible connection between curves T_i and T_j such that out of the two connecting points lying in T_i and T_j, at least one is a head or tail, and the other could be an interior point on the curve. If multiple connections are feasible between a pair of curves, then we retain only the nearest one that minimizes the separation between trajectories.

Table 1: List of features assigned to each connection in the multiple regression analysis.

	Feature	**Description**
1,2	Mean and Sum of absolute deviation of $K_{refined}$ on the interpolated track.	Each connection is composed of a track-head (or tail) and a section of another track at two ends. A cubic polynomial is fit to interpolate between these two end locations. Statistics computed from this interpolated track forms features.
3-5	Coefficients of cubic polynomial	
6	Cosine of angle of approach	The angle between the two track fragments forming the connection, such that when they are aligned, the angle is 0 and when they are perpendicular, the angle is 90 degrees.
7-9	Average $K_{refined}(\mathbf{u})$ of track retained, and discarded, and the ratio of these two quantities	If a connection is formed such that a track head is connected to another track in the middle, the other track gets split into two parts. One part that obeys the time-axis direction forms a part of the new track, and another part is discarded. The average $K_{refined}(\mathbf{u})$ values from the two parts, i.e., part of the track retained and the part that is discarded is used as feature.
10	Average $K_{refined}(\mathbf{u})$ of track	Average $K_{refined}(\mathbf{u})$ value computed on the complete track.
11	Ratio of Feature 10 and 7	Ratio of average $K_{refined}(\mathbf{u})$ value track retained to the complete track.
12	Ratio of Feature 10 and 8	Ratio of average $K_{refined}(\mathbf{u})$ value track discarded to the complete track.

We compute a list of features (shown in Table 1) corresponding to each feasible connection. Using multiple regression on the feature list, each connection is mapped to a scalar quantity that measures the validity of the connection. We used linear support vector regression to learn the weights of regression analysis from a sample training set that was manually scored by an expert. The scalar output of the regression analysis is then sorted in order of the connection validity and bad connections are discarded. If there is more than one connection possible for a tip, then the one with higher validity is retained. The qualified connections result in a merger of the corresponding curves forming complete granule trajectories. Lastly, we apply a minimum trajectory duration criteria of 1 second to filter out small fragmented trajectories that were due to artifacts and noise to obtain the final set of trajectories.

6. Experimental Results and Analysis

The video sequences for our experiments were collected using the protocol described in [5], [29]. Briefly, an expression construct for mCherry-tagged BDNF was transfected into mouse cortical neurons, derived from E18 embryos and cultured for 7 days. Co-expression of soluble EGFP was used to delineate neuronal morphology and identify areas where the orientation of the axon relative to the cell body was clear and the direction of transport (anterograde/retrograde) was evident. Cells grown on a coverslip were mounted in a closed and heated chamber and imaged by a Yokogawa CSU-10 spinning disk confocal microscope on a Nikon TE2000 inverted scope with continuous axial-drift compensation. For illumination, an AOTF-controlled laser line selection on a Coherent Innova 70C Krypton Argon Ion laser was used. Images were collected at ∼2Hz on a high-resolution Hamamatsu Orca ER CCD camera with resolution of 0.18 pixels per micron.

The algorithm was evaluated on a sample dataset containing 12 video sequences. Each video sequence was collected from an axon located in the "middle region" that was a few hundred microns from the cell body. The granule transport in this region is dominated by long-range movements. The granule motion was studied using the kymographs along with frame-by-frame examination of the video.

The algorithm parameters described in the previous sections are set as follows: The σ used in the tracing of the central axis is set proportional to the average width of the axon. In this study, the axons were about 4-8 pixels in width, and therefore, the value of σ is set to 5 pixels. The orientation distribution function was generated using a kernel of width $l = 12$ in (2). The voting neighborhood size was set to $N_b = 30$ pixels in (4). The stopping threshold b in Algorithm 1 was set as the median value of the $K_{refined}$ intensity. The joining step used a width of the search window that was set at 5 pixels. We ran all of our analysis programs on a personal computer running Windows XP OS with 4GB RAM in native MATLAB R2007a with a few modules coded in C++. The kymograph generation code ran in a few seconds, while the total time to process voting and trajectory detection modules was on the order of 1 minute depending on the kymograph size.

6.1. General Validation

We validated the performance of the algorithm against the judgment of an expert biologist for each of 12 kymographs, which we refer as ground truth. Figure 9 shows a typical result produced by the algorithm with trajectories overlaid on the kymograph in different colors. The biologist could review the original video to help determine the ground truth trajectories. The automated algorithm's results were compared to the ground truth and classified based on the amount by which the trajectories generated by the automated algorithm overlapped with the ground truth trajectories. A point on a ground truth trajectory is considered "covered" by an automatically detected trajectory if the automatically detected trajectory lies within one frame along the time axis and $1\mu m$ along the spatial axis of the ground truth trajectory on the kymograph.

The amount by which an automatically detected trajectory covered a ground truth trajectory is thus measured by a *coverage fraction*. At a given coverage fraction, each algorithm trajectory is classified as a hit or true positive (TP), i.e., a ground truth trajectory was covered by the specified fraction, or a false positive (FP), i.e., it did not cover any ground truth trajectory by the specified fraction. If a ground truth trajectory was covered by no algorithm trajectory, it was classified as miss or false negative (FN). Biologists can set a coverage fraction to characterize the performance of the algorithm; here, we varied this fraction from 30% to 90%.

For a fixed coverage fraction, the performance of the automated algorithm is determined by measuring the *recall rate* and *precision*. Recall rate is defined as the percentage of ground truth trajectories detected, i.e., TP/(TP + FN). Precision is defined as the percentage of algorithm trajectories corresponding to some ground truth trajectory, i.e., TP/(TP + FP).

Figure 10 shows the precision vs. recall of the automated detection process averaged over the 12 test sequences as the coverage fraction defining TP, FN, and FP was varied from 30% to 90%. Note that this is not a standard precision-recall curve in the sense of measuring performance as a function of an independent operating parameter. Instead, it measures precision and recall in terms of the coverage threshold; both measurements naturally decrease as the criterion for being a hit becomes more stringent.

The recall rate is satisfactorily high- above 89% in our experiments, falling gradually as the coverage fraction exceeds 70%. This means that almost all of the ground truth trajectories are covered by at least 70% of their apparent length in the kymograph, and that there are few misses at this level (around 6% of ground truth). The precision of the algorithm drops as the coverage fraction is raised, and the value is between 70% and 80% for high coverage fractions, indicating false positives rates of 20% to 30%. However, the situation is better than it appears.

Table 2: Performance of the algorithm averaged over the twelve video sequences for a coverage fraction of 70%. Recall and precision are reported with respect to trajectory duration and physical length.

Overall Recall	94.79%
Overall Precision	82.82%
Recall w.r.t. duration	97.61%
Precision w.r.t. duration	89.04%
Recall w.r.t. length	96.93%
Precision w.r.t. length	91.65%

In Table 2, we fix the coverage fraction to 70%, which corresponds to good overall levels of precision and recall, and analyze the results in more detail. When the duration (i.e., x axis extent in the kymograph) and physical length (i.e., y axis extent in the kymograph) of the trajectories in each category are taken into account, we see that most of the false positives are due to trajectories of short duration and physical length. Table 2 shows that while the average recall and precision at a coverage fraction of 70% are 95% and 83% respectively, these statistics improve to 98% and 89% when measured with respect to duration only. Similarly, the statistics improve to 97% and 92% when measured with respect to physical length only. Thus, we can conclude that the algorithm misses very few trajectories (high recall) and false positives mostly occur as short trajectories that has small extent in length and time. This implies that most of the long and prominent trajectories were successfully detected by the algorithm. This also implies that the algorithm's false positives and misses would have less influence on aggregate profiling of parameters that quantify the granule velocities within the axon, as described in the next section.

6.2. Oxidative stress experiment

Next, we describe a set of experiments used to quantify the effects of reactive oxygen species on secretory granule transport. In this experiment, axonal transport of granules was imaged before, and at varying times after adding peroxynitrite to the culture medium. The kymographs corresponding to the videos collected at 10, 25, 35 and 45 minutes intervals after the treatment are shown in Figure 11(a-d). Each kymograph spans about 78 seconds from the start of the image acquisition. The kymographs show that this reactive oxygen species causes a profound inhibition of granule transport and that anterograde transport is more susceptible to this treatment than retrograde transport. We applied our automatic algorithm to quantify these observations, which we term as trajectory profiling.

We first describe a few parameters that are used for validation and analysis of granule motion. First, we compute the instantaneous velocity of granules at each frame, as long as these granule also appear in the previous and subsequent video frames. The instantaneous velocities are computed using the central difference method from the neighboring video frames. In addition, several features are computed for each trajectory including the total duration of the trajectory, net displacement, and median velocity. The duration of the trajectory and net displacement are computed by measuring the horizontal and vertical extent of the trajectory in the kymograph. The median velocity for a trajectory is computed from the instantaneous velocities of the points comprising it. By convention, we denote anterograde velocity as positive and retrograde velocity as negative.

Normalized histograms of instantaneous velocity of trajectories are shown in Figure 12 at each treatment interval for the automatic algorithm on the left and the human expert on the right. The minimum resolution in velocity that determines the bin interval in the velocity histogram is given by the pixel dimension in the kymograph. The resolution in the horizontal direction is determined by the frame rate of the video, while in the vertical direction it is determined by the spatial sampling of the central axis (1 pixel in our case).

From Figure 12, it can be seen that for the video at 10 minutes of treatment, the number and velocity distribution of granules moving in the anterograde direction is almost identical to the number of granules moving the retrograde direction. However, for videos at 25 minutes, 35 minutes and 45 minutes after treatment, the percentage of movements in the anterograde direction clearly decreases. The histograms generated by the algorithm compare favorably with the results generated by the human expert. Except for a few errors due to histogram binning and pixel quantization, the overall trend in the velocity distribution is captured correctly by the proposed automated method. However, the automated algorithm is immensely faster, computing each histogram in only a few minutes.

Figure 13 reports several parameters including the total length of trajectories, the number of motion events and the anterograde-retrograde bias in the time lapse sequences in the form of histograms. The total length of each trajectory was computed by summing the net displacement of all granules in the movie. Similarly, the number of motion events gives a count of the number of granule movements during the movie. The first two parameters provide an indication of the overall amount of transport that is occurring in the axon. From the histograms, it can be seen that maximum transport is observed at 10 minutes, and the amount of transport gradually declined in the videos at 25, 35 and 45 minutes, which was corroborated by the expert's analysis. The algorithm also accurately detected the change in anterograde - retrograde bias, defined as the ratio of the amount of transport in the anterograde direction versus transport in the retrograde direction in the axon.

7. Conclusions

This study demonstrates the practicality of automated detection and analysis of BDNF granules moving along axons. The approach can be generally applied to any situation involving particles moving along 1D paths, such as, transport of mitochondria in axons, objects traveling along the blood stream inside blood vessels, or even road-traffic monitoring systems. The use of kymographs mirrors the practice in manual analysis of biological images, and provides important advantages for automated analysis as well as a convenient static approach for visualization. We believe our work is an important first step towards applying advances in image processing algorithm designs to kymograph analysis in axonal transport, a process that is extremely time consuming for human analysis.

The reconnection strategy adopted in our paper is simple compared to other reconnection schemes proposed in the literature (e.g.,[7]). In our ground-truth observations, BDNF granules rarely merged and split, so a simple trajectory-joining scheme was sufficient. Furthermore, since were interested in analyzing the aggregate velocity patterns of granule transport within the axon (e.g., Section 6.2), any sparse errors due to splits and merges were tolerable. Incorporating a full algorithm for dealing with splits and merges in more complex datasets would improve our algorithm's robustness.

The more ambitious task of extracting all trajectories is possible with our method, but with a lower level of automation. The automated algorithm described here produced very few missed trajectories, but there are several false positives due to noise and imaging artifacts. We noted that the intensity of stationary granules was always higher than that of the moving granules due to the blur caused by motion. Although this difficulty was partly addressed using the temporal sharpening step, it still remains challenging to detect all the granules in the data. Moreover, some of the slower granules moved erratically, and lacked regularity in velocity during motion. These granules were difficult to track and were at times detected as fragments of trajectories.

If the user desires a full extraction of all trajectories, the automated analysis results requires a moderate amount of editing, for which we have developed a graphical tool. In other words, our method is semi automated for this more ambitious task. In the future, we expect more complete automation to become possible, by combining advances in imaging instrumentation and protocols, and further improved algorithms that build upon the work presented here. Finally, we note that the axon extraction step can also be automated, e.g., by using an automated neuron tracer [30], creating kymographs and velocity profiles for each axon, and presenting the results to the user for further study.

8. Acknowledgments

This work was supported by grant ID A-2588 from Cure Huntington's Disease Initiative Inc. (CHDI Foundation), and by NIH grant MH 66179. The authors are thankful to Dr. Andrew Cohen (now at the University of Wisconsin, Milwaukee) for helpful discussions, and Dr. Stefanie Kaech-Petrie of the Jungers Center for Neurosciences Research, OHSU, Portland, for encouragement, criticism and advice. We would also like to extend our gratitude to Drs. Hyunsun Park and Richard Morse at the CHDI foundation for support and encouragement.

References

[1] C. Zuccato, E. Cattaneo, Brain-derived neurotrophic factor in neurodegenerative diseases, Nature Reviews Neurology 5 (6) (2009) 311–322.

[2] L. Gauthier, B. Charrin, M. Borrell-Pagès, J. Dompierre, H. Rangone, F. Cordelières, J. De Mey, M. MacDonald, V. Leßmann, S. Humbert, et al., Huntingtin controls neurotrophic support and survival of neurons by enhancing BDNF vesicular transport along microtubules, Cell 118 (1) (2004) 127–138.

[3] I. Ferrer, E. Goutan, C. Marin, M. Rey, T. Ribalta, Brain-derived neurotrophic factor in Huntington disease, Brain research 866 (1-2) (2000) 257–261.

[4] R. Barkus, O. Klyachko, D. Horiuchi, B. Dickson, W. Saxton, Identification of an axonal kinesin-3 motor for fast anterograde vesicle transport that facilitates retrograde transport of neuropeptides, Molecular biology of the cell 19 (1) (2008) 274–283.

[5] S. Kaech, G. Banker, Culturing hippocampal neurons, Nature protocols 1 (5) (2007) 2406–2415.

[6] R. T. Collins, Mean-shift blob tracking through scale space, in: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, Vol. 2, 2003. doi:10.1109/CVPR.2003.1211475.

[7] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. Schmid, G. Danuser, Robust single-particle tracking in live-cell time-lapse sequences, Nature methods 5 (8) (2008) 695–702.

[8] J. E. Boyd, J. Meloche, Y. Vardi, Statistical tracking in video traffic surveillance, in: Proc. Seventh IEEE Int Computer Vision Conf. The, Vol. 1, 1999, pp. 163–168. doi:10.1109/ICCV.1999.791213.

[9] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, IEEE Transactions on Signal Processing 50 (2) (2002) 174–188. doi:10.1109/78.978374.

[10] N. Ray, S. Acton, Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours, IEEE transactions on medical Imaging 23 (12) (2004) 1466–1478.

[11] D. Padfield, J. Rittscher, N. Thomas, B. Roysam, Spatio-temporal cell cycle phase analysis using level sets and fast marching methods, Medical Image Analysis 13 (1) (2009) 143–155.

[12] O. Welzel, D. Boening, A. Stroebel, U. Reulbach, J. Klingauf, J. Kornhuber, T. Groemer, Determination of axonal transport velocities via image cross-and autocorrelation, European Biophysics Journal 38 (7) (2009) 883–889.

[13] W. B. Ludington, W. F. Marshall, Automated analysis of intracellular motion using kymographs in 1, 2, and 3 dimensions, Vol. 7184, SPIE, 2009, p. 71840Y. doi:10.1117/12.812419.
URL `http://link.aip.org/link/?PSI/7184/71840Y/1`

[14] I. Smal, I. Grigoriev, A. Akhmanova, W. Niessen, E. Meijering, Microtubule dynamics analysis using kymographs and variable-rate particle filters, Image Processing, IEEE Transactions on 19 (7) (2010) 1861–1876.

[15] E. Meijering, M. Jacob, J. Sarria, P. Steiner, H. Hirling, M. Unser, Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images, Cytometry Part A 58 (2) (2004) 167–176.

[16] T. Stepanova, I. Smal, J. van Haren, U. Akinci, Z. Liu, M. Miedema, R. Limpens, M. van Ham, M. van der Reijden, R. Poot, F. Grosveld, M. M., E. Meijering, N. Galjart, History-Dependent Catastrophes Regulate Axonal Microtubule Behavior, Current Biology 20 (11) (2010) 1023–1028.

[17] A. Frangi, W. Niessen, K. Vincken, M. Viergever, Muliscale Vessel Enhancement Filtering, in: Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer-Verlag, 1998, pp. 130–137.

[18] J. N. Tsitsiklis, Efficient algorithms for globally optimal trajectories, IEEE Transactions on Automatic Control 40 (9) (1995) 1528–1538. doi:10.1109/9.412624.

[19] J. Sethian, A fast marching level set method for monotonically advancing fronts, Proceedings of the National Academy of Sciences of the United States of America 93 (4) (1996) 1591–1595.

[20] C. Steger, An unbiased detector of curvilinear structures, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (2) (1998) 113–125.

[21] D. Geman, B. Jedynak, An active testing model for tracking roads in satellite images, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1) (1996) 1–14.

[22] M. Sofka, C. Stewart, Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures, IEEE Transactions on medical imaging 25 (12) (2006) 1531–1546.

[23] B. Parvin, Q. Yang, J. Han, H. Chang, B. Rydberg, M. H. Barcellos-Hoff, Iterative voting for inference of structural saliency and characterization of subcellular events, IEEE Transactions on Image Processing 16 (3) (2007) 615–623. doi:10.1109/TIP.2007.891154.

[24] M. Nicolescu, G. Medioni, Motion segmentation with accurate boundaries - a tensor voting approach, in: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, Vol. 1, 2003. doi:10.1109/CVPR.2003.1211379.

[25] P. Kornprobst, G. Medioni, Tracking segmented objects using tensor voting, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Vol. 2, 2000, pp. 118–125. doi:10.1109/CVPR.2000.854756.

[26] G. Medioni, M. Lee, C. Tang, A computational framework for segmentation and grouping, Elsevier Science Ltd, 2000.

[27] O. Monga, S. Benayoun, Using partial derivatives of 3d images to extract typical surface features, in: Proc. Third Annual Conf AI, Simulation and Planning in High Autonomy Systems Integrating Perception, Planning and Action. of, 1992, pp. 225–236. doi:10.1109/AIHAS.1992.636890.

[28] P. Felzenszwalb, D. McAllester, A min-cover approach for finding salient curves, in: Proc. Conf. Computer Vision and Pattern Recognition Workshop CVPRW '06, 2006. doi:10.1109/CVPRW.2006.18.

[29] S. Kaech, C. Huang, G. Banker, Live Imaging of Developing Hippocampal Neurons in Culture, in: Imaging in Developmental Biology: A Laboratory Manual (R. Wong and J. Sharpe, eds.), 2011.

[30] O. Al-Kofahi, R. Radke, S. Goderie, Q. Shen, S. Temple, B. Roysam, Automated cell lineage tracing: a high-throughput method to analyze cell proliferative behavior developed using mouse neural stem cells, Cell Cycle 5 (3) (2006) 327–335.

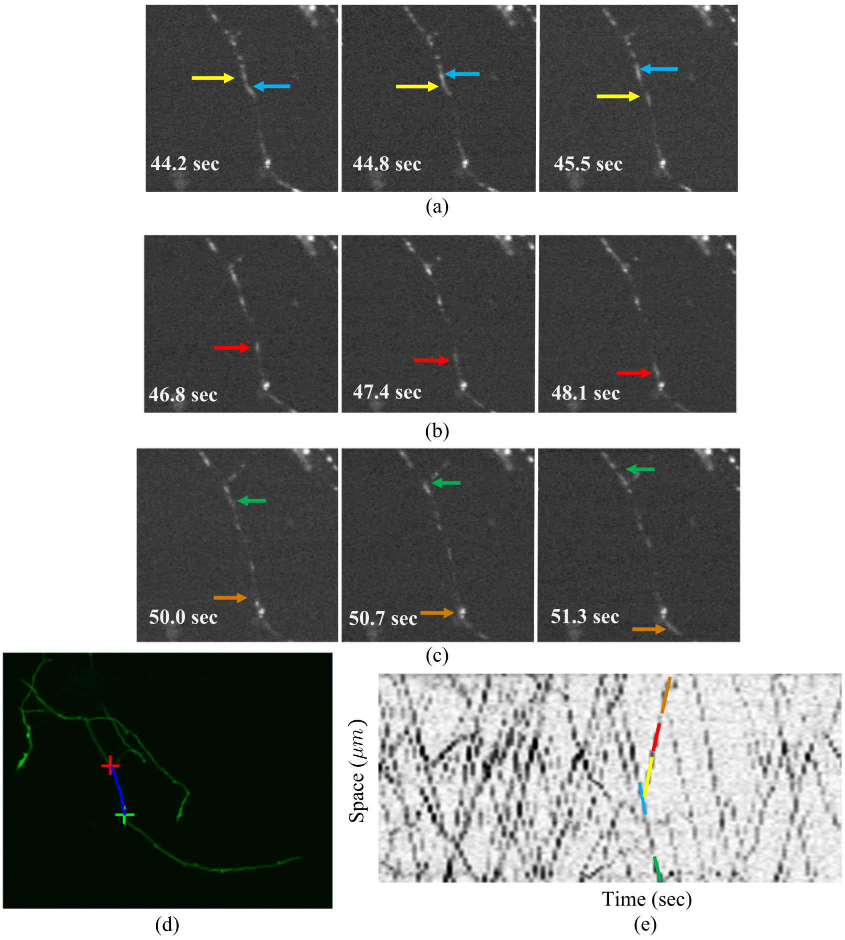


Figure 1: (a-c) Frames from a video ($18\mu m \times 18\mu m$ region at 1.8 fps) along mouse cortical neurons showing transport of BDNF granules. (a) Intersecting granule trajectories (b) changing morphology and (c) passing a bright stationary granule. (d) Section of the eGFP labeled axonal image selected for analysis. The spatial resolution of the image is $0.18\mu m$/pixel. (e) A space-time map (kymograph) of the video showing the same granule motion. (This figure is best viewed in color)

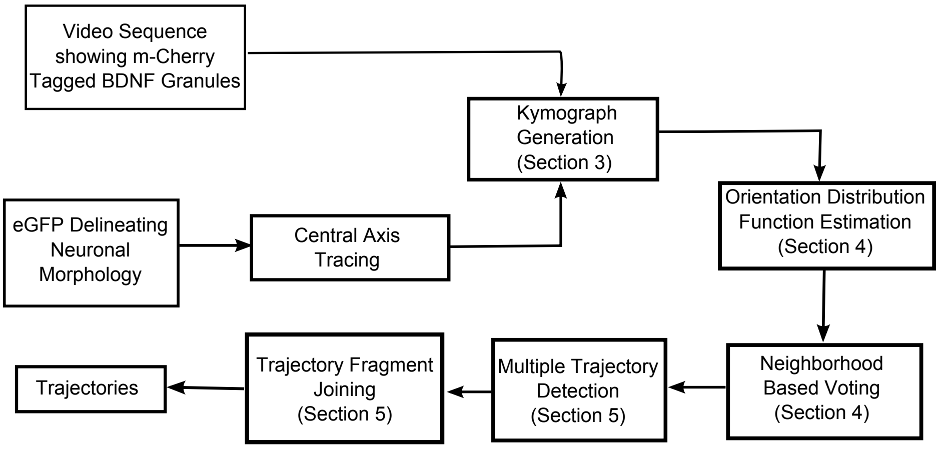
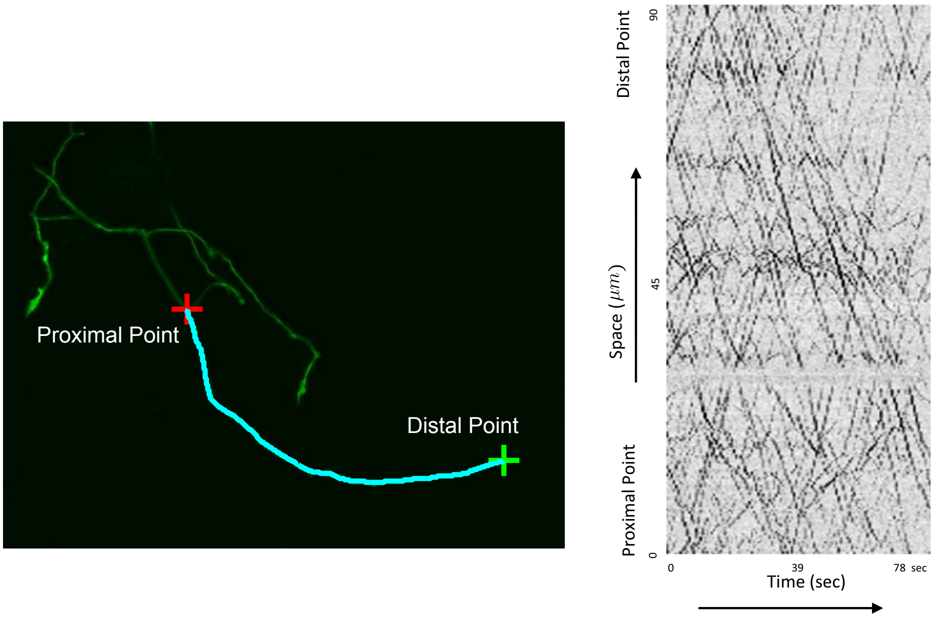

Video Sequence showing m-Cherry Tagged BDNF Granules
Kymograph Generation (Section 3)
Orientation Distribution Function Estimation (Section 4)
eGFP Delineating Neuronal Morphology
Central Axis Tracing
Neighborhood Based Voting (Section 4)
Multiple Trajectory Detection (Section 5)
Trajectory Fragment Joining (Section 5)
Trajectories

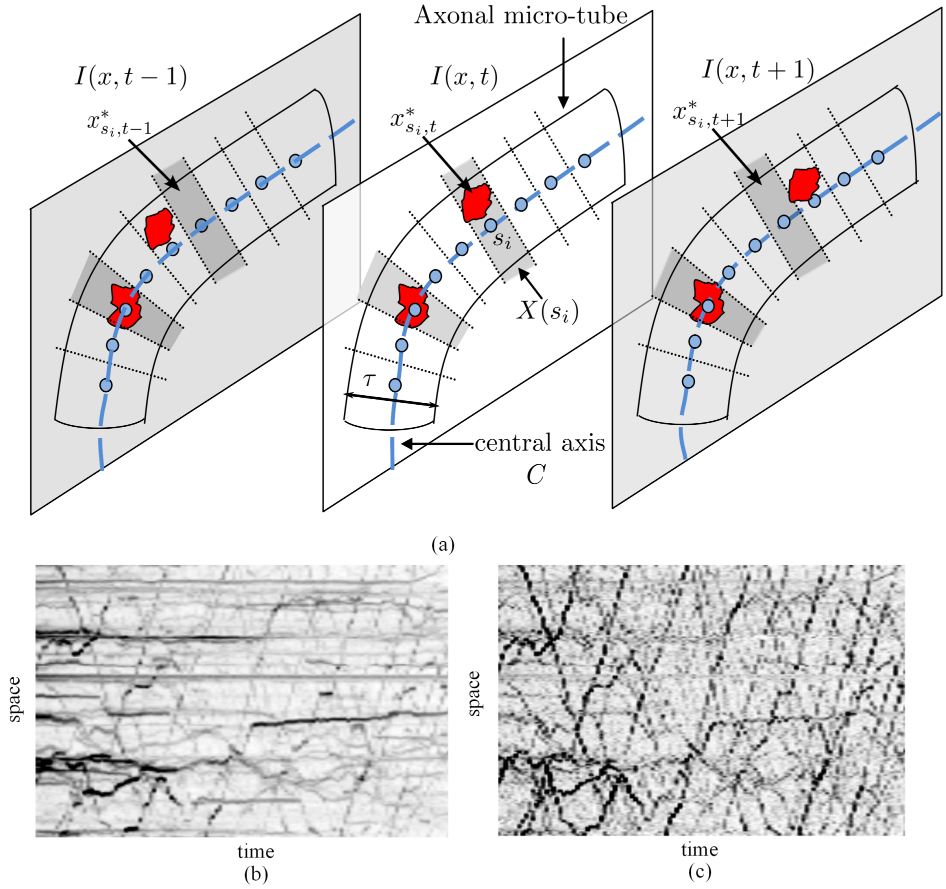
Figure 2: Flowchart showing steps in the proposed algorithm for velocity profiling of Brain Derived Neurotrophic Factor (BDNF) granules.



(a)

(b)

Figure 3: (a) A section of the eGFP labeled axon that is selected for BDNF granule transport analysis. The proximal and distal points on the axon are chosen by the user. The estimated central axis is shown as a blue curve. The spatial resolution of the image is $0.18\mu m/pixel$. (b) The kymograph generated from this section. The horizontal axis of the kymograph indicates time, such that the left boundary indicates start time while the right boundary indicates end time. The vertical axis of the kymograph indicates space, where the top and bottom correspond to the distal and the proximal points of the axonal section, respectively.

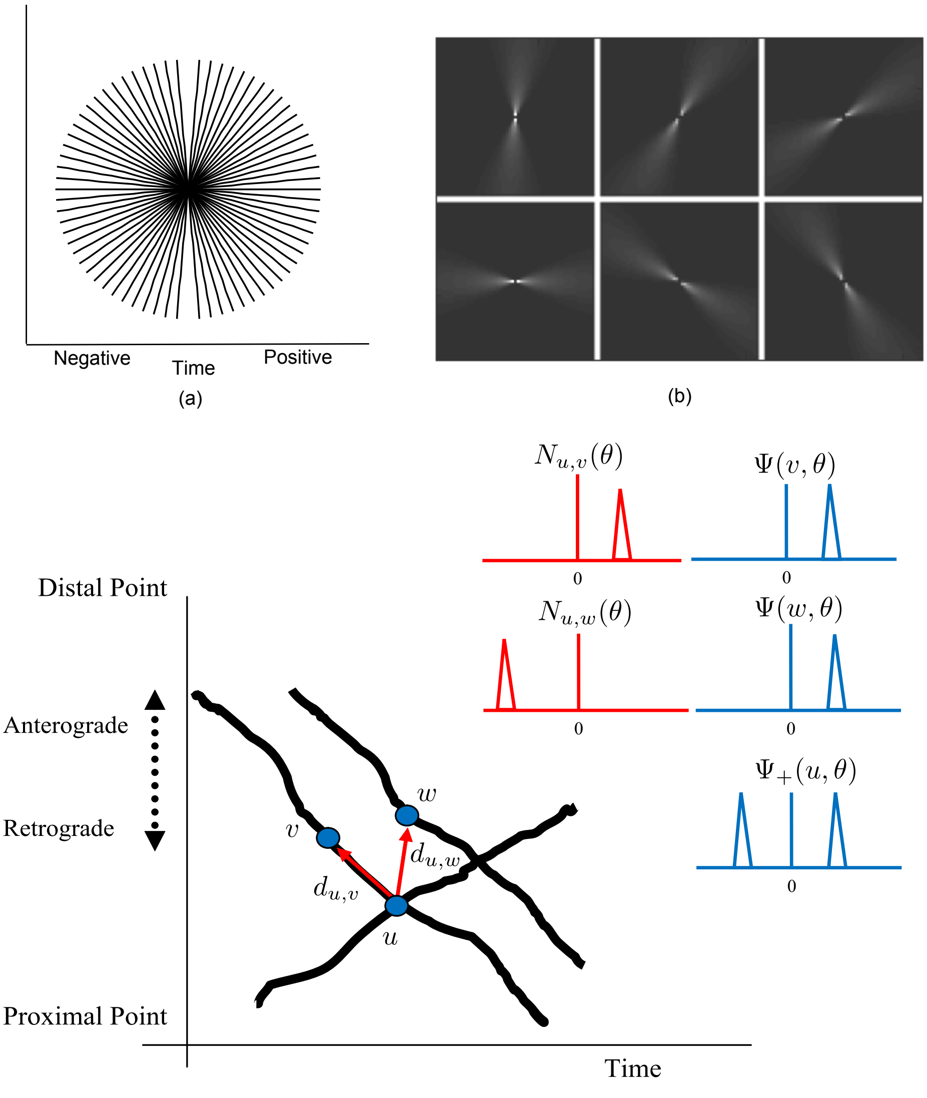


(a)

(b)

(c)

Figure 4: (a) The kymograph is generated by parameterizing the central axis of the axonal-tube by s_i. The region $X(s_i)$ is illustrated over three consecutive time frames. Two representative granules are shown in red: one of them is moving and the other is stationary. The dilation width τ for generating the kymograph are specified by the user. (b) Kymograph generated before temporal sharpening. (c) Kymograph after temporal sharpening.



(a)

(b)

(c)

Figure 5: Illustrating the orientation distribution function and the voting process. (a) The ODF kernel showing the directions of line integration. (b) The neighborhood decay functions at several selected orientations, using the parameters $m = -0.1$ and $\sigma = 15\,\text{deg}$. (c) A space-time pixel u is shown with two neighbors v and w. The relative positions of v and w with respect to u are indicated by the vector $d_{u,v}$ and $d_{u,w}$. The voting process compares the ODFs at u with those of its neighbors, weighted by the decay function to generate votes.

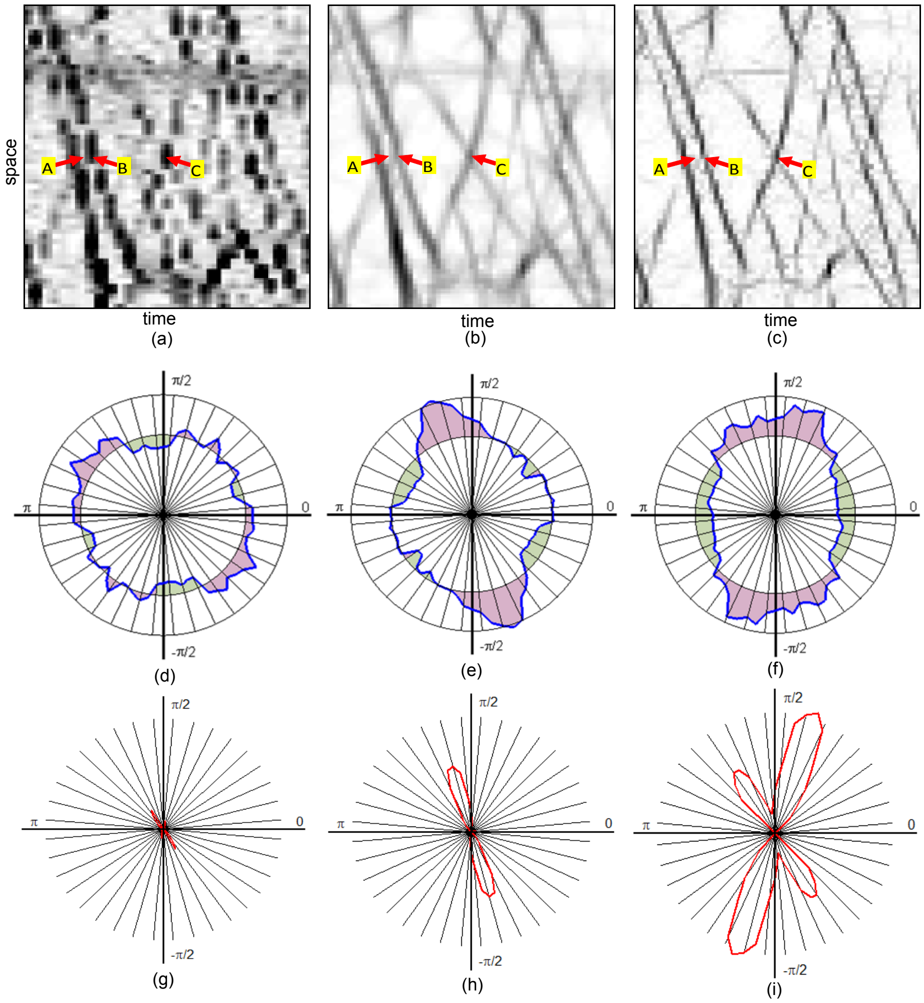


Figure 6: Illustrating the voting process. (a) A spatio-temporal region in a kymograph with 3 sample points indicated by A, B and C. (b) The result of the voting operation (normalized for visualization) (c) The response of refining operation. (d-f) The ODF kernel responses on a polar graph at points A, B and C respectively before voting (g-i) ODFs at the same locations after voting.

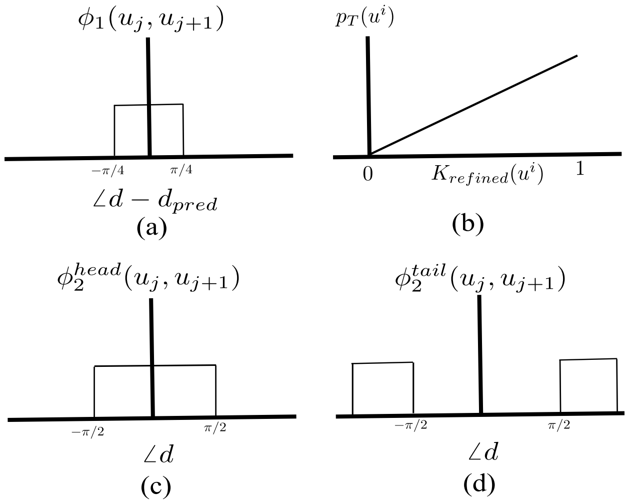


Figure 7: (a) The function ϕ_1 for implementing the smoothness constraint. (b) The function p_T that relates the cost to the value of $K_{refined}$, (c,d) The functions ϕ_2^{head} and ϕ_2^{tail} for implementing the feasibility constraint for curve heads and tails.

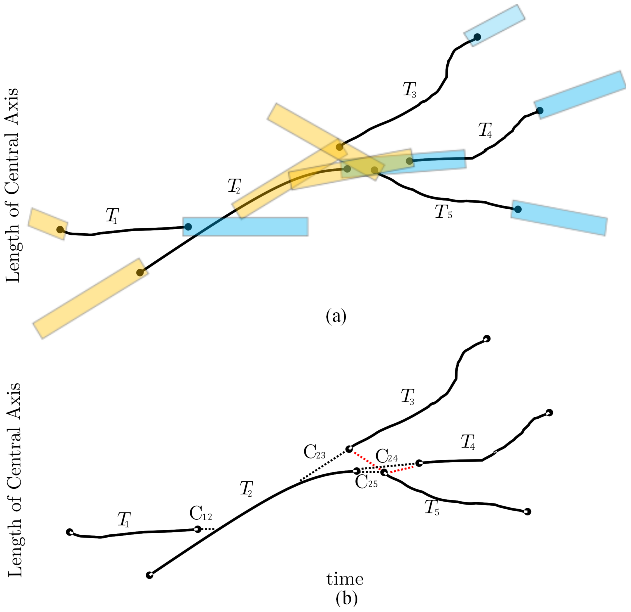


Figure 8: (a) The search window for each head (blue) and tail (yellow). (b) All possible connections that can be made between the tips and other trajectories. Connections shown in red are not possible due to the feasibility constraint. The final joining selects a subset of these connections to determine the final trajectories.

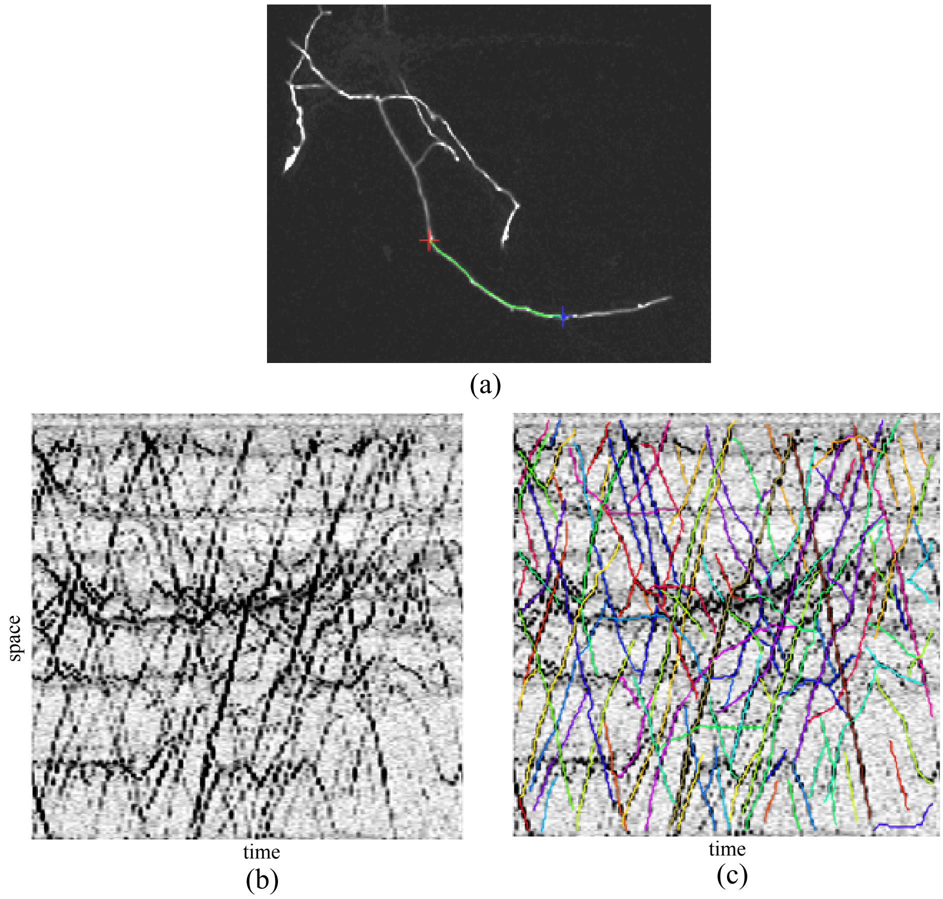


Figure 9: (a) A section of the axon shown in Figure 3 selected for analysis; (b) The automatically generated kymograph between the proximal and distal points shown in Panel (a); (c) Automatically detected trajectories detected on the kymograph. Each trajectory is assigned a random color for visualization.

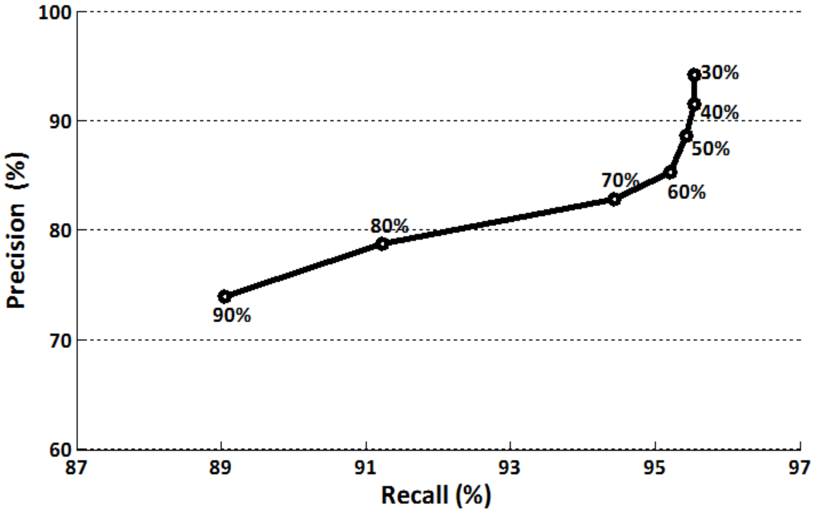


Figure 10: Precision vs. recall of the automated algorithm, averaged over the twelve video sequences. Each point represents a value of the coverage fraction.

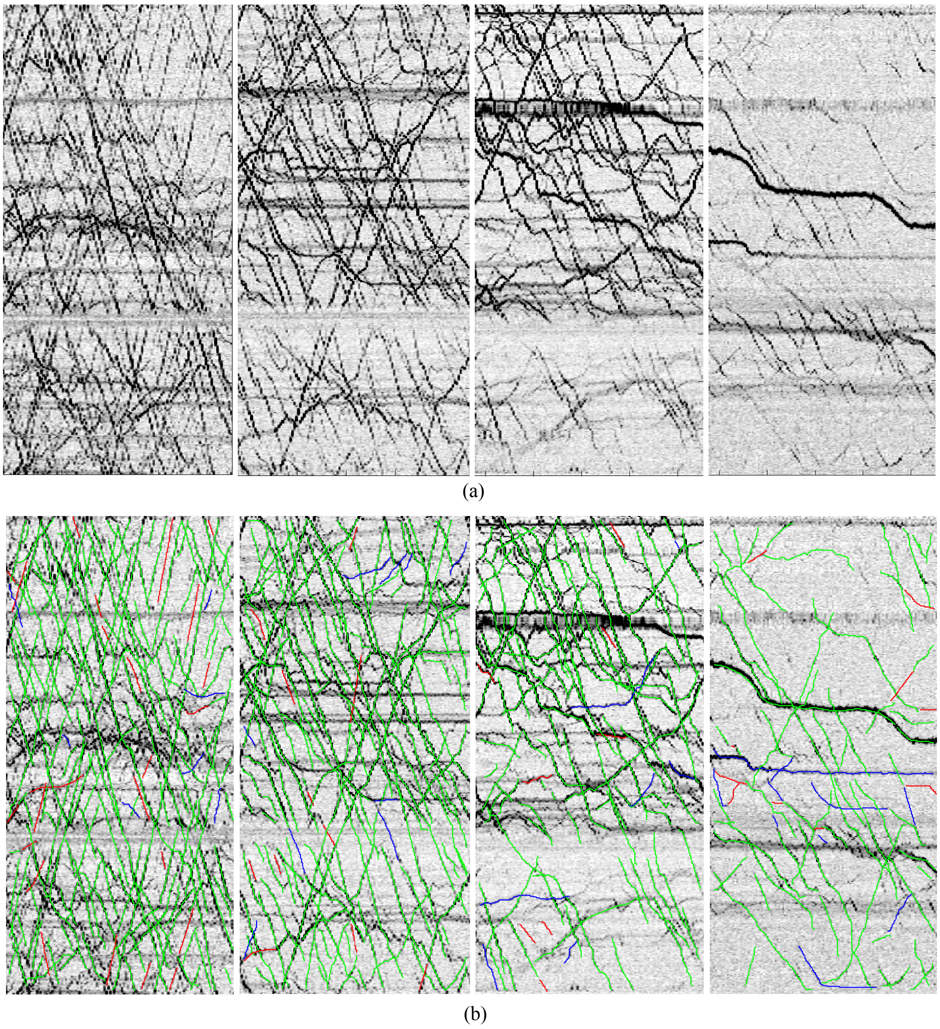


Figure 11: Kymographs and results corresponding to the videos used in experimental evaluation and analysis of the algorithm. (a) Kymographs computed from videos acquired at 10 minutes, 25 minutes, 35 minutes and 45 minutes after the treatment. (b) The trajectories after recovery - green lines show the trajectories that were correctly picked by the automated algorithm, red shows the misses while blue lines indicate false positives.

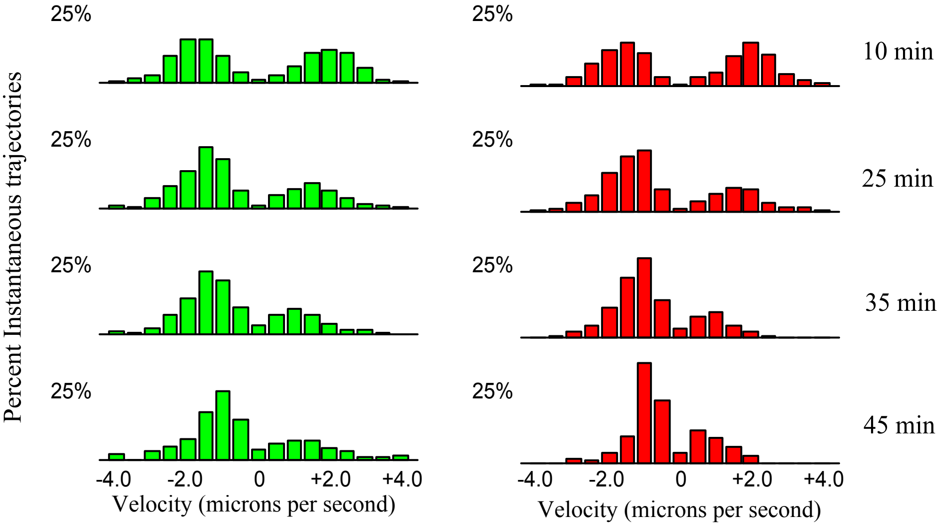


Figure 12: Histograms of instantaneous trajectories at acquisition times of 10 minutes, 25 minutes, 35 minutes and 45 minutes after the treatment respectively. The horizontal axis of each histogram shows the velocity of instantaneous trajectory, while the vertical axis gives the count of number of trajectory with a given velocity. Positive velocity denotes anterograde motion, whereas negative velocity denotes retrograde motion. The left column shows results for automated analysis, while the right column shows results of analysis by an expert.

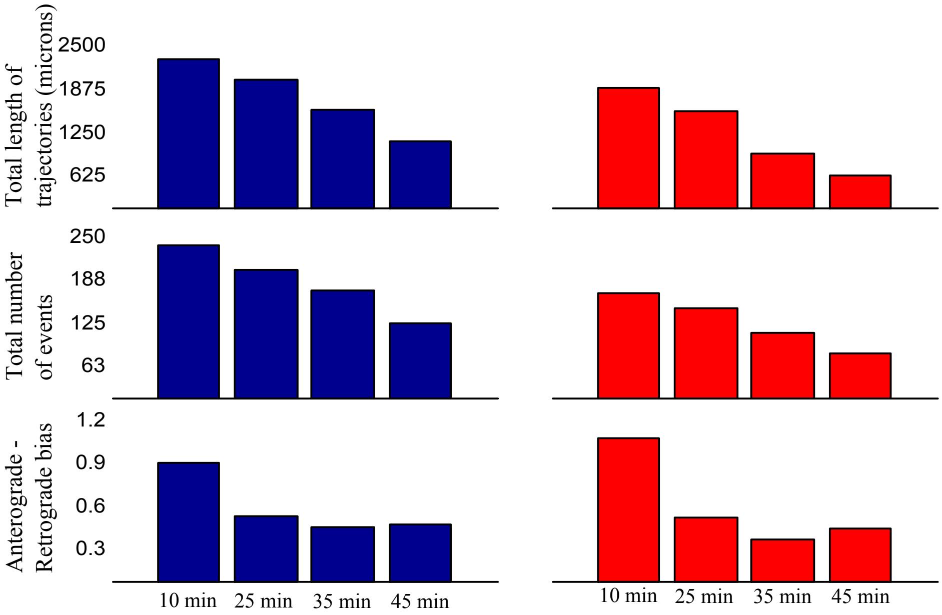


Figure 13: Trajectory profiling parameters. (Top row) Total length of trajectories in microns as found by the automated algorithm (left) and by the expert (right). (Middle row) Total number of motion events and (Bottom row) Anterograde-Retrograde bias.