Automated Analysis of Kymograph Images for Quantitative Study of Ciliary Transport

by

Srijita Chakraburty

(Under the Direction of Suchendra Bhandarkar)

Abstract

Intraflagellar transport (IFT) is essential for the construction of cilia and flagella in eukaryotic cells and plays a critical role in the transport of protein cargo to and from the cell body. Protein molecules involved in IFT are observed to travel at different velocities. These velocities are computed using spatio-temporal maps called kymographs. Kymographs are single images that represent the 3D microscopy images of intracellular motion as 2D time series data. In vivo microscopy imaging typically results in the generation of noisy kymographs which are difficult to analyze. Existing techniques for IFT velocity measurement entail manual detection of IFT trails on the kymographs, followed by computation of the IFT velocities via determination the slope of each IFT trail. Since manual kymograph analysis is laborious, time consuming and error prone, an automated algorithm to extract IFT trails and determine the IFT velocities in kymographs with minimal manual intervention is a valuable tool for biologists. To this end, we propose a machine learning-based approach to the analysis of kymographs that segments and delineates the IFT trails, and computes the IFT protein velocities. In the proposed approach, the kymograph image is preprocessed to suppress noise and identify potential IFT trail pixels. The potential IFT trail pixels are further characterized using Gabor wavelet transform (GWT) and curvelet transform (CT) features. A Support Vector Machine (SVM) is used to classify the potential IFT trail pixels into three IFT trail categories followed by the extraction of continuous IFT trajectories and the computation of the IFT velocity associated with each IFT trajectory. Experimental results show the advantages of the proposed approach in terms of accuracy and greatly reduced processing time for kymograph analysis.

INDEX WORDS: Kymographs, Intraflagellar transport (IFT), Support Vector Machines, Classification, Gabor Features, Curvelet Features

Automated Analysis of Kymograph Images for Quantitative Study of Ciliary Transport

by

Srijita Chakraburty

B.Tech., West Bengal University of Technology, India, 2011

A Thesis Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Master of Science

Athens, Georgia

2016

Automated Analysis of Kymograph Images for Quantitative Study of Ciliary Transport

by

Srijita Chakraburty

Automated Analysis of Kymograph Images for Quantitative Study of Ciliary Transport

Srijita Chakraburty

December 9, 2016

Acknowledgments

I would first like to thank my thesis advisor Dr. Suchendra Bhandarkar . He consistently steered me in the right direction whenever he thought I needed it. I would also like to thank Dr. Karl Lechtreck for providing me with all the data that contributed towards my research. I would also like to acknowledge all the members of Visual and Parallel Computing Laboratory. Their passionate participation and input was very valuable all throughout.

Finally, I must express my very profound gratitude to my parents and to my spouse for providing me with unfailing support and continuous encouragement through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Contents

1 INTRODUCTION **1**

2 LITERATURE REVIEW **6**

3 PROPOSED SYSTEM FOR AUTOMATED KYMOGRAPH ANALYSIS **9**

3.1 SYSTEM OVERVIEW 9

3.1.1 FEATURE EXTRACTION 10

3.1.2 CLASSIFICATION 16

3.1.3 POST-PROCESSING 23

3.1.4 VELOCITY-CALCULATION 24

4 RESULTS **25**

5 SUMMARY AND CONCLUSIONS **37**

List of Figures

1.1 Intraflagellar Transport in Flagella and Cilia 2

1.2 Comparison of kymographs obtained using two fluorescent markers: (a) mNeonGreen-IFT54 marker and (b) green fluorescent protein (GFP) marker. 3

3.1 System Overview 10

3.2 (a) An ensemble of Gabor wavelets (1.5 octave bandwidth). (b) Gabor wavelet coverage of the spatial frequency plane. Each ellipse shows the half-amplitude bandwidth contour dilated by a factor of 2, covering almost the complete support of a wavelet. 12

3.3 (a) The magnitude of the GWT representation. (b) The phase of the GWT representation. 13

3.4 Curvelet transform frequency Tiling (a):Continuous Domain. (b): Discrete Domain 14

3.5 Edge Representations 15

3.6 SVM hyperplane separating the data points. 17

3.7 Continuous OAA 18

3.8 Feature Image Creation-Gabor Features for anterograde trails 19

3.9 Feature Image Creation-Curvelet Features for retrograde trails 20

3.10 Post Segmentation Results using the Curvelet Transform (a) Raw Kymograph (b) Anterograde Trails (c) Retrograde Trails 22

3.11 Regression analysis 24

4.1 Post Classification Results Using the GWT (a) Raw Kymograph (b) Anterograde Trails (c) Retrograde Trails 26

4.2 Post Classification Results Using the CT (a) Raw Kymograph (b) Anterograde Trails (c) Retrograde Trails 26

4.3 Post Classification Results (a) Orientation Measure (b) Distance Measure . . 27

4.4 After Post-processing and Line-fitting is Performed on the Gabor SVM results (a) Raw Kymograph (b) Retrograde (c) Anterograde 27

4.5 (a) Velocity for each identified anterograde trail(Gabor SVM) (b) Velocity for each identified retrograde trail(GaborSVM) 30

4.6 (a) Velocity for each identified anterograde trail(Curvelet SVM) (b) Velocity for each identified retrograde trail(Curvelet SVM) 30

4.7 (a) Histogram showing velocity measure GWT-SVM detected trails: Anterograde (b) Histogram showing velocity measure GWT-SVM detected trails: Retrograde 31

4.8 (a) Histogram showing velocity measure CT-SVM detected trails: Anterograde (b) Histogram showing velocity measure CT-SVM detected trails: Retrograde 31

4.9 Average Anterograde Particle Velocity of Ten Kymographs(GWT-SVM) (a) Blue bar represents manually calculated velocity (b) Yellow bar represents manually calculated velocity 32

4.10 Average Anterograde Particle Velocity of Ten Kymographs(GWT-SVM) (a) Blue bar represents manually calculated velocity (b) Yellow bar represents manually calculated velocity 32

4.11 Average Anterograde Particle Velocity of Ten Kymographs(CT-SVM) (a) Blue bar represents manually calculated velocity (b) Yellow bar represents manually calculated velocity 33
4.12 Average Retrograde Particle Velocity of Ten Kymographs(CT-SVM) (a) Blue bar represents manually calculated velocity (b) Yellow bar represents manually calculated velocity 33
4.13 (a) Raw Kymograph (b) Probability Map: LDA classifier (c) Retrograde Trails: Gabor SVM Classifier+ Post- processing(Proposed Method) (d) Anterograde Trails: Gabor SVM Classifier+ Post- processing(Proposed Method) (e) Retrograde Trails: Curvelet SVM Classifier(Proposed Method) (f) Anterograde Trails: Curvelet SVM Classifier(Proposed Method) 34
4.14 (a) Gabor SVM (b) Curvelet SVM 34
4.15 [b] Raw kymograph from [1] [a] shows result of our proposed method: Gabor SVM on [1] [c] shows result of [b]. The red lines depict the lines detected by the algorithms and the blue lines show the missed out lines. 35
4.16 [b] Raw kymograph from [1] [a] shows result of our proposed method: Curvelet SVM on [1] [c] shows result of [b]. The red lines depict the lines detected by the algorithms and the blue lines show the missed out lines. 35
4.17 [a] Raw kymograph [b]Hough tranform to detect anterograde trails. Red lines are the trails detected [c]. The green lines depict the anterograde lines detected by our algorithm: GWT-based SVM before post processing. 36
4.18 (a) Weak Kymograph(Raw) (b) Anterograde Trails: Gabor SVM Classifier(Proposed Method) (c) Retrograde Trails: Gabor SVM Classifier(Proposed Method) . 36
4.19 (a) Weak Kymograph(Raw) (b) Probability Map: LDA classifier 36

List of Tables

4.1 Confusion Matrix After Classification: GWT-based SVM 28
4.2 Confusion Matrix After Classification: CT-based SVM 28
4.3 Number of Trails Detected by Each Method 28
4.4 Average Localization Error: In Terms of Number of Pixels 28

Chapter 1

INTRODUCTION

Flagella and cilia are appendages or organelles that protrude from the cell body of certain prokaryotic and eukaryotic cells and are associated with cell motility and cellular sensory functions [2]. Flagella and cilia are known to be sensitive to the chemical composition and temperature of the extracellular environment and perform important physiological roles in chemical sensation, signal transduction, and control of cell growth [3]. In the current scientific understanding, flagella and cilia are sensory cellular antennae that coordinate a large number of cellular signaling pathways that control the division, motility and differentiation of cells [4].

The inner core of cilia and flagella comprises of a microtubule-based cytoskeletal structure called the axoneme. The axonemal cytoskeleton provides a scaffolding for molecular motor proteins that enable intraflagellar transport (IFT), a process by which proteins are conveyed up and down the microtubules [5]. IFT describes the bi-directional movement of protein particles, that are made up of several individual proteins, along the doublet microtubules of the flagellar axoneme. IFT proteins are primarily concentrated at the base of the flagellum or cilium. Anterograde transport denotes the movement of the IFT protein particles away from the cell body and towards the tip of the flagellum or cilium where they are assembled, whereas retrograde transport represents movement of the protein particles, that represent

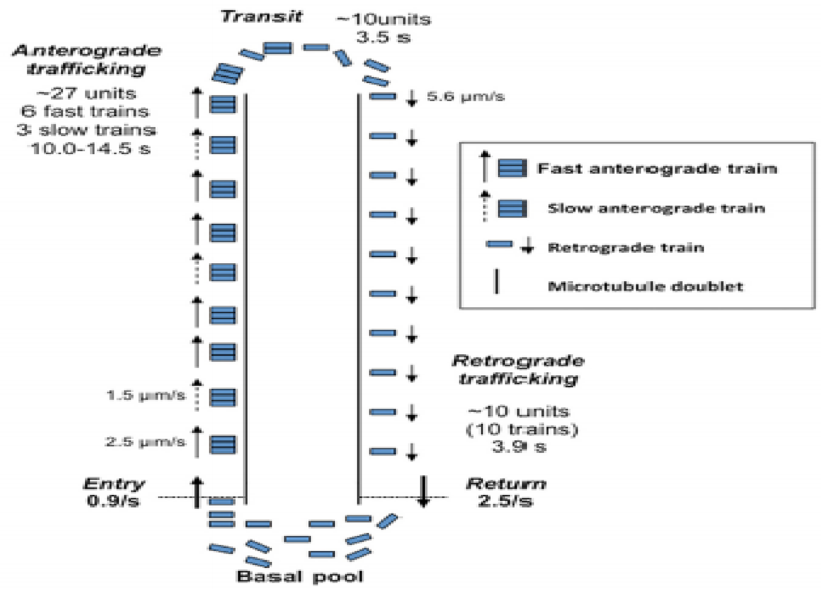

Figure 1.1: Intraflagellar Transport in Flagella and Cilia

turnover products, from the tip of the flagellum or cilium back towards the cell body or cell center.

Other than assisting in cargo transport to and from the cell body, the IFT machinery is also critical for the formation and maintenance of a healthy flagellum or cilium. In fact, an axoneme with defective IFT machinery is observed to slowly shrink in the absence of replacement protein subunits. On account of its importance in genesis and maintenance of functional cilia and flagella, defective IFT machinery has been implicated in several diseases associated with malfunctioning, non-functioning or absent cilia (i.e., ciliopathy) such as polycystic kidney disease [6], polycystic liver disease [7], congenital heart disease [8] and retinal degeneration [9], among others.

Usually there are several IFT protein molecules moving along the microtubules of the flagellar axoneme resulting in the generation of multiple protein particle trajectories along the same path. A formal study of the dynamics of these protein particles is of particular importance to cellular biologists, since it allows for quantification the velocity, frequency and volume of protein transport between the cell body and its environment. Time lapse microscopy (TLM) is a commonly used technique for dynamic imaging of the IFT phenomenon. The TLM videos are typically captured at a rate of 10 frames/second (fps) and magnification of 9.2 pixels/micron using a Total Internal Reflection Fluorescence (TIRF) microscope [10]. The IFT proteins are infused with a fluorescent marker to improve their visibility and enable their tracking in the TLM video with greater ease and accuracy. There have been many different types of fluorescent markers described in the research literature; with the green fluorescent protein (GFP) being one of the most commonly used [11]. However, the GFP marker has been observed to result in weak or low-intensity IFT protein trails in the TLM videos, prompting the recent popularity of the mNeonGreen-IFT54 fluorescent marker [12] which has been observed to result in more prominent trails.

The TLM video stream is converted to a kymograph which is a single static image that provides a 2D graphical representation of the spatial position of a particle, moving along a well defined path, over time. In our case, the particle of interest is the IFT protein particle and the well defined path is the axis of the flagellum or cilium. One of the axes (the y-axis in our case) in the kymograph represents the spatial displacement of the particle along the

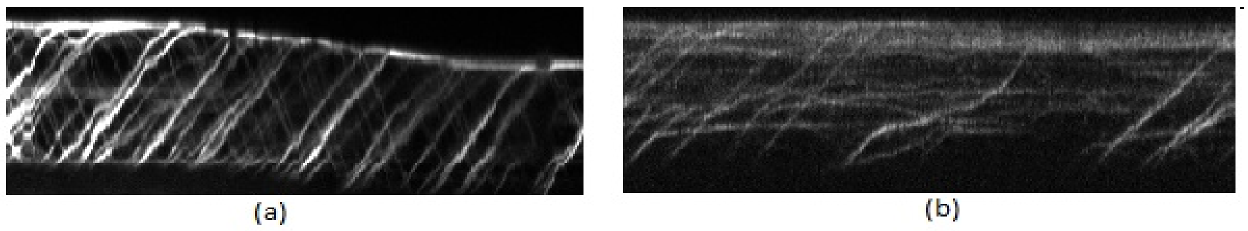

Figure 1.2: Comparison of kymographs obtained using two fluorescent markers: (a) mNeonGreen-IFT54 marker and (b) green fluorescent protein (GFP) marker.

axis of the flagellum or cilium whereas the other axis (i.e, the x-axis) in the kymograph represents time. Figure 1.2 shows two kymographs; one obtained using the mNeonGreen-IFT54 marker and other using the GFP marker. As is evident, the trails in the kymograph obtained using the mNeonGreen-IFT54 marker are more prominent than the ones in the kymograph obtained using the GFP marker. The slanted trails (with non-zero slope values) in the kymograph represent moving particles whereas stationary particles are denoted by horizontal trails in the kymograph. The slope of a trail in the kymograph represents the velocity at which the corresponding IFT protein particle moves along the axis of the flagellum or cilium.

In this paper we propose an automated method to extract multiple IFT protein particle trajectories from a single kymograph image. These trajectories are observed to be along both directions of the flagellar or ciliary axis; i.e., anterograde and retrograde [13]. In the proposed method we aim for reliable and automated tracking of multiple IFT protein particles in the presence of several cross-points in the kymograph space. Cross-points are points where anterograde and retrograde trails cross over each other. There is a general paucity of automated methods for kymograph analysis in the biological research literature. Moreover, there have been very few instances of machine learning-based techniques used for automated kymograph analysis.

In this paper we exploit the Gabor wavelet transform (GWT) [14] and the curvelet transform (CT) [15] (which is a generalization of the wavelet transform) to characterize the pixels potentially belonging to the trails of IFT protein particles experiencing anterograde and retrograde motion. A support vector machine (SVM) is trained for classifying these trails into three distinct classes namely *anterograde motion*, *retrograde motion* and *background* (which includes stationary IFT particles, IFT particles exhibiting random motion on account of diffusion and noisy artifacts) followed by the computation of anterograde and retrograde velocities of the corresponding IFT particles.

In summary, the pipeline of the proposed method comprises of three main components: (a) prepossessing of the noisy kymograph and extraction of potential IFT trail pixels (b) characterization of the potential IFT trail pixels using the GWT and CT features (c) training of an SVM-based classifier to categorize the extracted IFT trail pixels into the aforementioned three distinct classes (d) SVM- based classification of extracted IFT trail pixels in the test image (e) computer-assisted extraction of spatially contiguous IFT particle trajectories from the classified trails and computation of the velocity of each of these trajectories.

The outline of the remainder of the paper is as follows: In Chapter 2 we review the state-of-the-art methods for automated analysis of kymographs. In Chapter 3 we provide a detailed description of the proposed method. We present and discuss the experimental results in Chapter 4. Finally, in Chapter 5, we present the conclusions with an outline of the directions for future work.

Chapter 2

LITERATURE REVIEW

Manual analysis of kymographs is not only labor intensive but also requires a biologist's expertise for proper interpretation of the kymograph data. The significance of automating the process of kymograph analysis has been realized by the research community in cellular biology. Manual kymograph analysis has been observed to be error prone or at least subjective; in that weaker kymograph trails could be identified as valid IFT particle trajectories or not depending on the observer. The goal of automation is to put kymograph analysis on a strict objective basis.

In recent years there have been several methods proposed in the research literature to automate the analysis of kymographs. However, most algorithms in the literature have not proven to be very effective when dealing with noisy kymographs generated from *in vivo* TLM videos. The TLM videos are captured at a relatively fast rate of 10 fps which results in the creation of kymographs with very low signal to noise ratio (SNR). As mentioned previously, the kymographs, in general, encapsulate both, anterograde and retrograde transport. Anterograde IFT particles are typically larger (i.e., consist of more proteins) and travel slower compared to the retrograde IFT particles. Consequently, anterograde trials on the kymographs are typically more prominent than the weaker and sometimes hard to discern retrograde trails.

The ImageJ plugin described in [16] is a Radon transform-based algorithm for automated kymograph analysis [17]. The Radon transform is generalization of the Hough transform [18] which is a popular technique to detect lines or linear features in an image. This approach works well when detecting prominent and non-fragmented straight lines along a single direction of the image. However, the kymographs we need to analyze contain particles displaying bidirectional movement which leads to the presence of several faint lines criss-crossing one another. This plugin fails to work on kymographs of such nature. The automated IFT velocity measurement technique proposed by Welzel et al. [19], which also uses the Hough transform to detect relevant IFT trails, is observed to work well on noise-free kymographs with very well defined IFT trails but is seen to be ineffective for analysis of low-SNR kymographs for the same reasons as the ImageJ plugin [16]. Nair et al. [20] present a kymograph analysis technique wherein a linear discriminant analysis (LDA)-based classifier is used to generate a probability map to delineate the IFT trails. The generated probability map gives the probability of each image pixel belonging to a trail but does not classify the pixel as belonging to an anterograde or retrograde trail. LDA is a generative classifier and is more suitable for binary classification (i.e., *trail* vs. *non-trail*), whereas our goal is to separate the trails into different classes based on their direction. Thus, this technique fails to detect most of the retrograde IFT trails in a low-SNR kymograph dataset such as ours.

A very recent paper by Mangeol et al. [21] describes the working of two tools; the first tool called KymographClear is used generate kymographs from the TLM videos whereas the second tool called KymographDirect is used to analyze these kymographs. The drawback of the analysis tool, KymographDirect, is that it lacks generality and can only be used to analyze kymographs generated by the first tool, KymographClear. In fact, KymographDirect is observed to not work on most existing kymographs generated using other techniques such as the ImageJ plugin [16]. The KymogaphDirect tool also failed to detect many trails due to extensive processing of the kymograph image by KymographClear, which resulted in the removal of many relevant trails from the raw kymograph image.

The method proposed by Mukherjee et al. [22] is a semi-automated technique for IFT trail detection that is based on a voting algorithm. Their technique is not well equipped for analysis of kymographs that have a high trail density arising from dense IFT particle traffic. Since their technique relies on a pixel neighborhood-based voting mechanism, many irrelevant trails that are essentially background noise, are misclassified as IFT trails. This limits the capability of their technique to analyze kymographs that contain noisy artifacts due to the presence of immobile protein particles and particles undergoing diffusion. The techniques based on Steger's curvilinear edge detection algorithm described in [1] and [23] are also observed not to work well on kymographs with IFT trails displaying low intensity values and noisy artifacts. The techniques described in these papers lay emphasis on a seed-based, region growing method. In situations where the IFT trails in the kymograph are very close to one another, two or more IFT trails are labeled as a single trail on account of the region growing algorithm. If the seeds for the relevant IFT trails are not initialized properly, the resulting detected trails are often noisy.

The proposed automated kymograph analysis technique presented in this paper addresses the primary challenges faced by cellular biologists, i.e., (a) identification of IFT trails corresponding to anterograde and retrograde transport in the presence of noisy artifacts arising from immobile protein particles and particles undergoing diffusion, (b) separation of anterograde and retrograde trails, (c) reduction in time taken to analyze kymograph images and (d) elimination of edge distortion introduced during manual tracing of IFT trails.

Chapter 3

PROPOSED SYSTEM FOR AUTOMATED KYMOGRAPH ANALYSIS

3.1 SYSTEM OVERVIEW

The proposed system, depicted in Figure 3.1, consists of a pipeline for segmentation and classification of kymograph images into three different types of trails: (a) *anterograde* trails, (b) *retrograde* trails and, (c) *stationary* trails and/or *background* folllowed by the computation of the particle velocities for the anterograde and retrograde trails. The ImageJ package is first used to generate kymographs from the TLM video streams. The kymograph images are generated with time along the x-axis and distance along the y-axis. The kymograph images are subject to contrast enhancement to increase the visibility of faint trails followed by segmentation and delineation of potential IFT trail pixels. The potential IFT trail pixels within the enhanced kymograph images characterized using the Gabor wavelet transform (GWT) and curvelet transform (CT) features. The extracted features in manually labeled kymograph images are used to train a support vector machine (SVM)-based classifier to classify the potential IFT trail pixels into the three aforementioned categories. During the testing phase, the segmented kymograph image is fed to the SVM classifier and the potential IFT pixels are classified into the three aforementioned categories. In the subsequent postprocessing phase, the results of the classification are used to extract continuous IFT trajectories in the kymograph image and compute the velocity associated with each IFT trajectory in the anterograde and retrograde category.

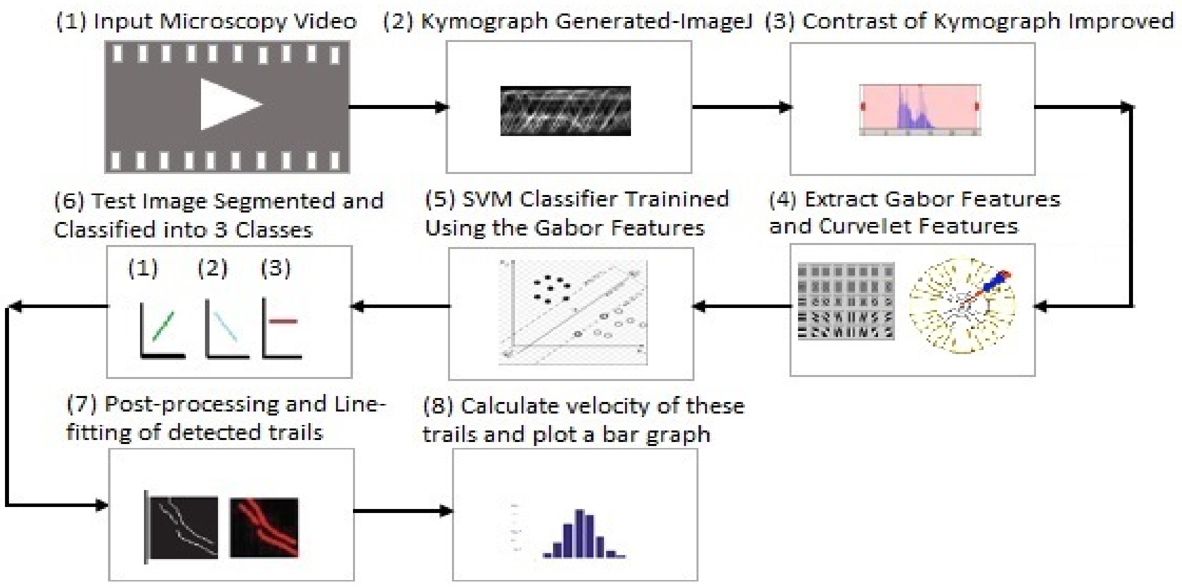
(1) Input Microscopy Video
(2) Kymograph Generated-ImageJ
(3) Contrast of Kymograph Improved
(4) Extract Gabor Features and Curvelet Features
(5) SVM Classifier Trainined Using the Gabor Features
(6) Test Image Segmented and Classified into 3 Classes
(1)
(2)
(3)
(7) Post-processing and Line-fitting of detected trails
(8) Calculate velocity of these trails and plot a bar graph

Figure 3.1: System Overview

3.1.1 FEATURE EXTRACTION

Feature extraction is the most important step in any automated classification system. In the proposed system, two different types of features, i.e., the Gabor wavelet transform (GWT) and the Curvelet transform (CT), are used to characterize the potential IFT trail pixels. These features are used to model two SVM-based classifiers and the results compared to existing machine learning techniques from the research literature.

GABOR WAVELET TRANSFORM

The Gabor wavelet transform (GWT), which captures the properties of orientation selectivity, spatial localization and optimal localization in the space and frequency domains, has been extensively and successfully used in various computer vision and image processing applications [14]. The frequency characteristics and orientation representations of the GWT have been observed to be quite similar to those of human visual system and have been found to be well suited for texture representation and texture discrimination. The GWT-based features directly from the gray-level images has been successful and widely been applied to texture segmentation. The default GWT kernel is given by:

$$\psi = \frac{k_{\mu,v}}{\sigma^2} e^{(-k_{\mu,v}^2 z^2/2\sigma^2)} [e^{ik_{\mu,v^z}} - e^{-\sigma^2/2}] \tag{3.1}$$

Where: μ and v define the orientation and the scale of the Gabor filters, z = (x, y) and $k_{\mu,v}$ is defined as following form:

$$k_{\mu,v} = k_v e^{i\phi u} \tag{3.2}$$

where, $k_v = kmax/f^v$ and $\phi u = \pi\mu/8$, kmax is the maximum frequency, and f is the spacing factor between kernels in the frequency domain. Usually $\sigma = 2\pi$, kmax $= \pi/2$ and $f = \sqrt{2}$. In this paper, $\mu \in$ [0, 1... ,7] and $v \in$ [1, 2, 3, 4].

The Gabor wavelet representation of a kymograph image is obtained by performing a convolution between the image and a family of Gabor filters as described by Eq(3). The convolution of image I(z) and a Gabor filter $\psi_{\mu,v(z)}$ can be defined as follows:

$$F_{\mu,v(z)} = I(z) * \psi_{\mu,v_z} \tag{3.3}$$

Where z = (x, y), * denotes the convolution operator, and $F_{\mu,v(z)}$ is the Gabor filter response of the image with orientation μ and scale v.

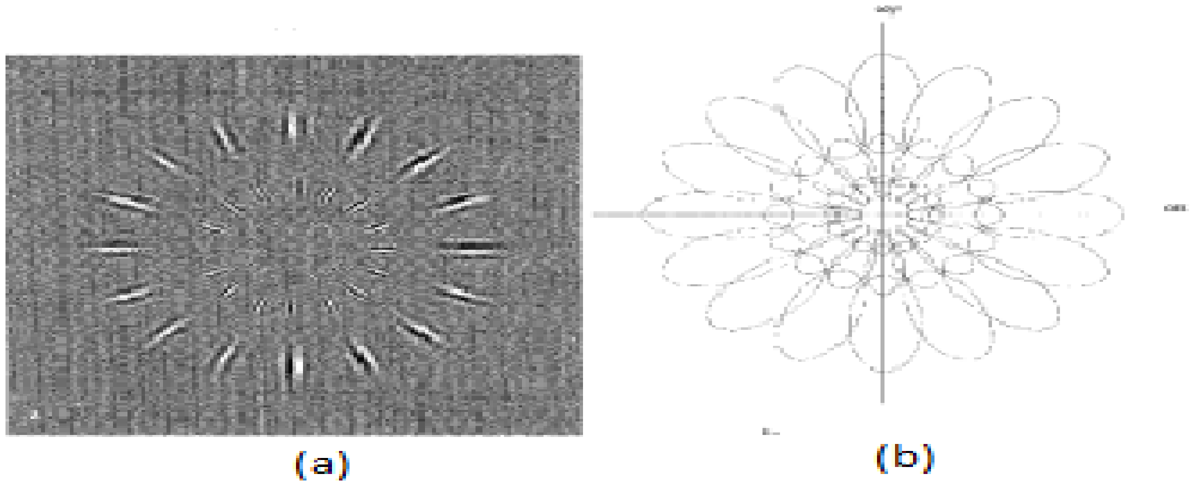

Figure 3.2: (a) An ensemble of Gabor wavelets (1.5 octave bandwidth). (b) Gabor wavelet coverage of the spatial frequency plane. Each ellipse shows the half-amplitude bandwidth contour dilated by a factor of 2, covering almost the complete support of a wavelet.

In this work, the number of orientations of the GWT is set to 8 and the number of scales is set to 5. The GWT kernel is oriented at two different directions for extracting the trails in the two different directions. An orientation of 45° is chosen for the anterograde trails and an orientation of 135° is chosen for the retrograde trails.

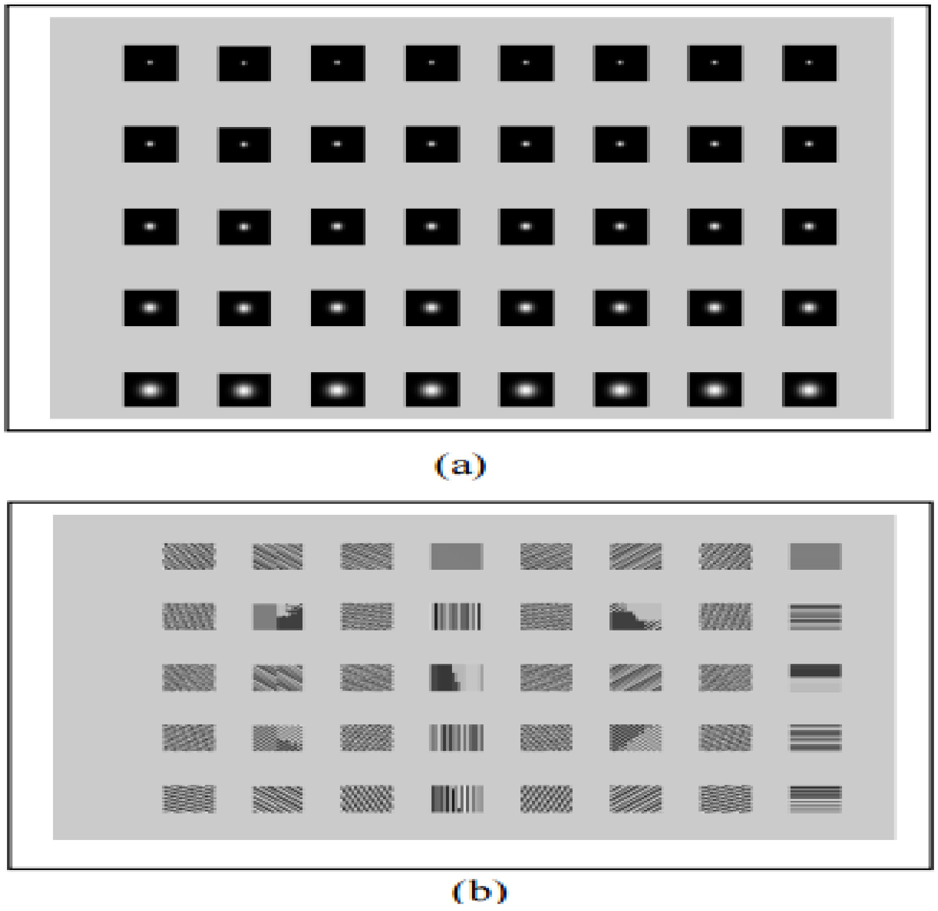

Figure 3.3: (a) The magnitude of the GWT representation. (b) The phase of the GWT representation.

CURVELET TRANSFORM

The basic idea behind the Curvelet transform (CT) is to represent a curve as a superposition of functions of various lengths and widths obeying the scaling parabolic law: width $\cong$ (length)^2. Figure 3.4 shows the CT frequency tiling which called the Second Dyadic Decomposition (SDD). The length of the localizing windows (colored blue) is doubled at every other dyadic sub-band. The CT in continuous domain is defined using coronae and rotations as shown in Figure 3.4(a). Since the discrete input data is defined on a Cartesian grid, the

Curvelet transform in the discrete domain is more conveniently described using concentric squares and shears instead of concentric circles and rotations, as shown in Figure 3.4(b). The frequency plane is partitioned using radial (circles and squares) and angular (rotations and shears) divisions. Different scales are obtained by radial division; the smallest scale defines the finest resolution while the largest scale defines the coarsest resolution. Angular division divides each scale into different orientation; the maximum number of orientations was found at the finest resolution and the lesser number of orientations was found at coarsest resolution.

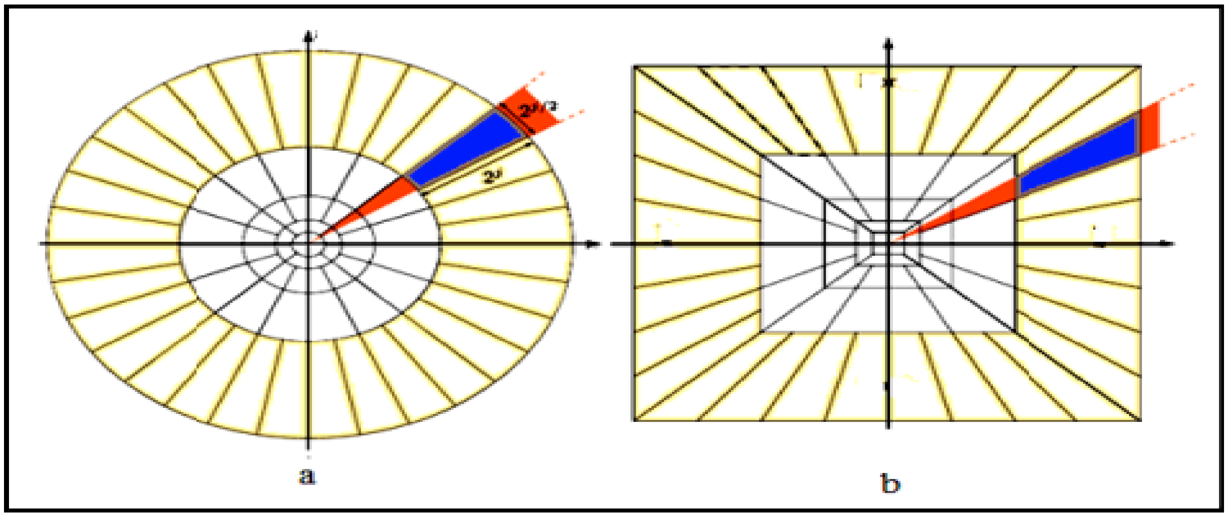

Figure 3.4: Curvelet transform frequency Tiling (a):Continuous Domain. (b): Discrete Domain

The CT can be regarded as an extension of the traditional wavelet transform. The CT is designed to represent edges and other singularities along curves much more efficiently than the traditional wavelet transform which good at representing point singularities. Figure 3.5 shows edge representation of a curve obtained using the traditional wavelet transform and the CT. As can be noted, it takes several wavelet coefficients to accurately represent such a curve whereas the CT needs much fewer coefficients, i.e, the wavelet transform needs three, six, and twelve coefficients, whereas the CT needs one, two, and four coefficients for the largest, intermediate, and smallest values of the scale parameter, respectively.

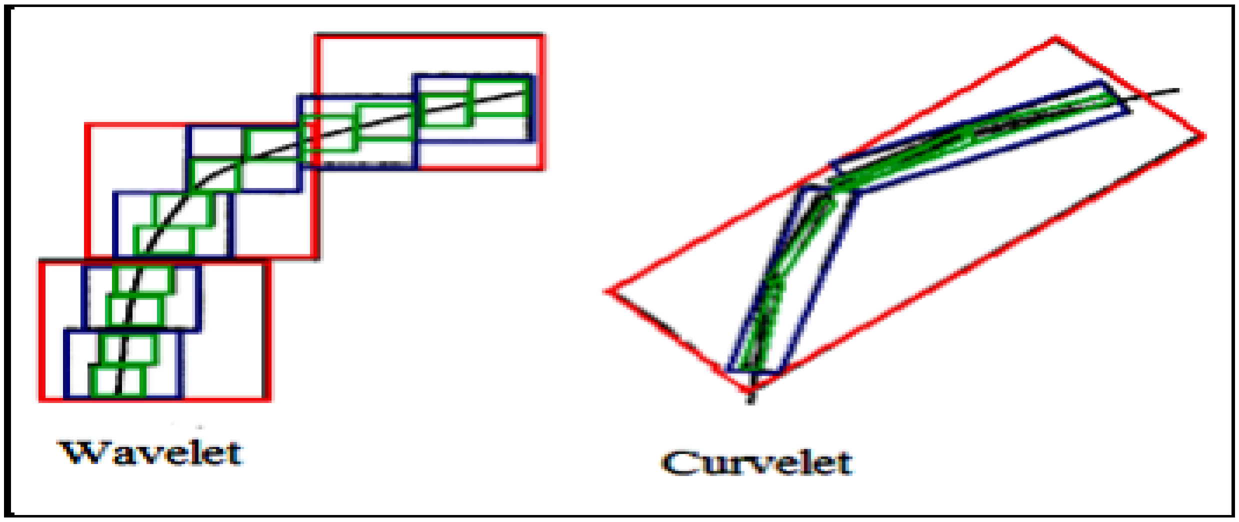

Figure 3.5: Edge Representations

The wrapping implementation of the curvelet transform [7] relies on the computation of the coefficient $c_{(j,l,k)}$ in the Fourier domain as:

$$c_{(i,j,k)} = \int f(\omega) U_j(S_{\Theta_l}^{-1}) e^{i<b,\omega>} d\omega \tag{3.4}$$

where $U_j(S_{\Theta_l}^{-1})$) is a smooth frequency window which is supported on a parallelepipedal region, where the matrix S_{Θ_l} is a shear matrix of angle Θ_l and $b \simeq (k_1 2^{-j}, k_2 2^{-j/2})$ with $k = (k_1, k_2)$. In practice, the frequency domain is tiled with a set of oriented smooth windows $U_{j,l}$ called wedges. The wedge-based decomposition results in an coronization of the frequency domain based on concentric squares and shears which are slightly overlapping as shown in Figure 3.4(a).

3.1.2 CLASSIFICATION

SUPPORT VECTOR MACHINE

The classifier used in this approach is the Support Vector Machine (SVM).The SVM is based on statistical learning theory and aims to determine the locations of linear decision boundaries that produce the optimal separation of classes [24]. The SVM generates a hyperplane between two sets of data for classification. In the case of a two-class pattern recognition problem where the classes are linearly separable, the SVM selects from among the infinite number of linear decision boundaries the one that minimizes the generalization error. Thus, the selected decision boundary will be one that leaves the greatest margin between the two classes, where margin is defined as the sum of the distances to the hyperplane from the closest points (or feature vectors) corresponding to the samples from the two classes [24]. This problem of maximizing the margin can be solved using standard Quadratic Programming (QP) optimization techniques. The data points or feature vectors that are closest to the hyperplane are used to measure the margin; hence these data points are termed as *support vectors*. Consequently, the number of support vectors is small compared to the total number of data points [24]. Thus the SVM can be regarded as a learning machine that classifies data by shaping a set of support vectors [25].

If the two classes under consideration are not linearly separable, the SVM tries to find the hyperplane that maximizes the margin while, at the same time, minimizing a quantity proportional to the number of misclassification errors. The trade-off between the margin maximization and misclassification error minimization is controlled by a user-defined constant [24]. The SVM can also be extended to handle non-linear decision surfaces. Boser et al. [26] propose a method for projecting the input data onto a high-dimensional feature space using kernel functions [24] and formulating a linear classification problem in that high-dimensional feature space. A primary benefit of the SVM is the low expected probability of generalization errors [27]. Moreover, once the data is classified into two classes, an appropriate optimizing algorithm can be used if needed for feature identification, depending on the application [24].

The data is mainly linear and is represented in a discriminative manner, which justifies our choice of using an SVM classifier. The feature vectors are of high dimensionality and SVMs are known to work better than most classifiers when dealing with feature vectors of higher dimensionality

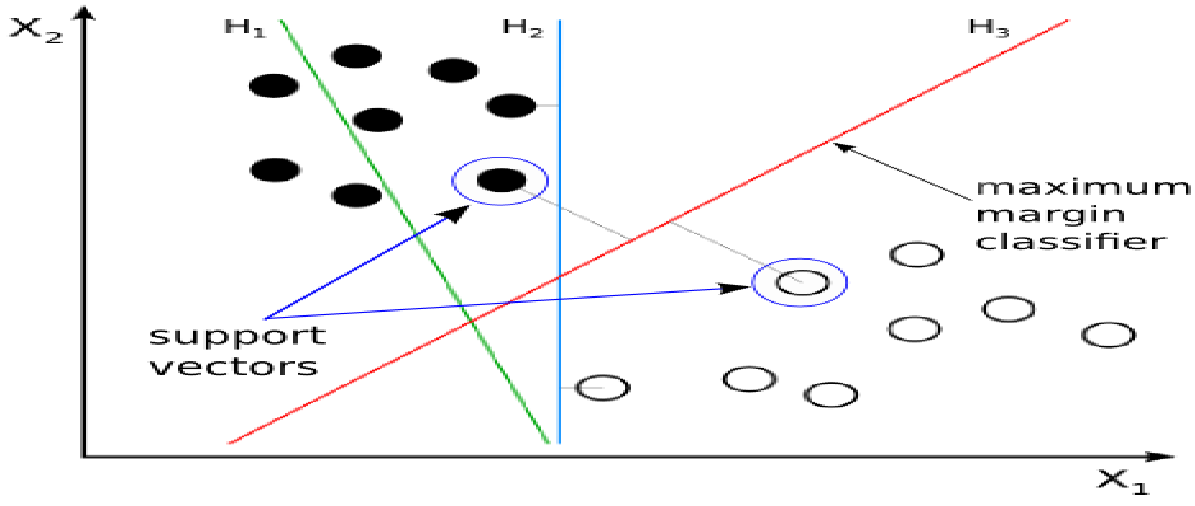

Figure 3.6: SVM hyperplane separating the data points.

The SVM was initially designed for binary (two-class) problems. When dealing with multiple classes, an appropriate multi-class method is needed. Vapnik [24] has suggested a one-against-all strategy based on comparing one class with all the other classes taken together. This strategy generates n classifiers, where n is the number of classes. The final output is the class that corresponds to the SVM with the largest margin, as defined above. For a multi-class problem one has to determine n hyperplanes. Thus, the one-against-all(OAA) strategy requires the solution of n QP optimization problems, each of which separates one class from the remaining classes. Since the aim of the proposed approach involves classifying into 3 classes, the problem is one of multi-class classification. Multi-class classification problems (where $n > 2$) are commonly decomposed into a series of binary problems such that the standard SVM can be directly applied. The one-against-all strategy is applied for classification where to get an n-class classifier, we construct a set of binary classifiers $f_1, f_2, \ldots, f_n$, each trained to separate one class from rest and then combine them to obtain multi-class classification.

The initial formulation of the one-against-all method required unanimity among all SVMs: a data point would be classified under a certain class if and only if that class's SVM accepted it and all other classes' SVMs rejected it. While accurate for tightly clustered classes, this method leaves regions of the feature space undecided where more than one class accepts or all classes reject. We have implemented a continuous OAA. It involves using the continuous values of SVM decision functions rather than simply their signs. The class of a data point is whichever class has a decision function with highest value, regardless of sign. This appears to be the most common method for multiclass SVM classification in use today.

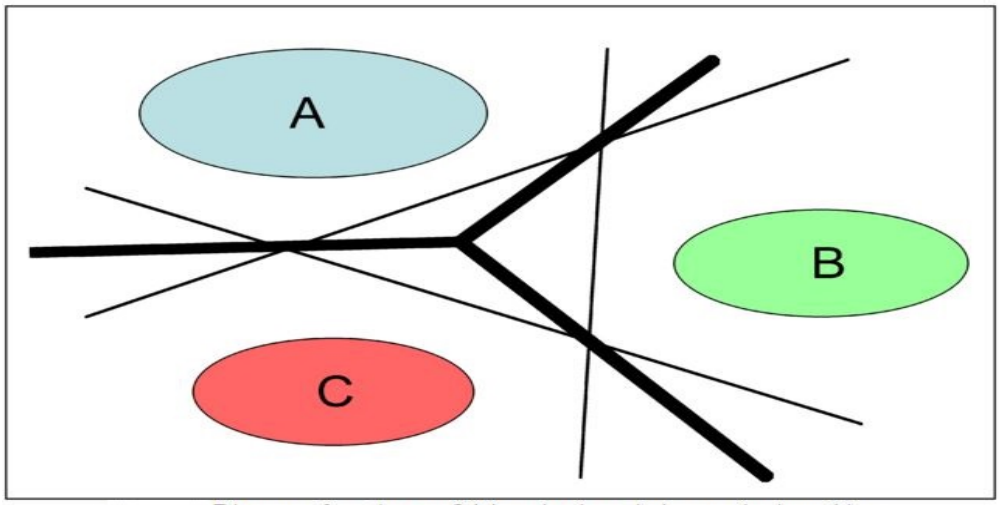

Diagram of continuous OAA region boundaries on a basic problem

Figure 3.7: Continuous OAA

TRAINING PROCEDURE

For training the SVM model, we first specify the type of features we want to use to build the model. All slanted lines aligned between 100° - 170° are marked as retrograde trails, all slanted lines aligned between 10° - 80° are marked as anterograde trails, all horizontal trails which represent stationary particles and vertical trails which could result from a flash applied during making the videos, are marked as negative instances along with anything that can be classified as background noise. The feature images for training are derived in the following manner: We first hand-draw the trails for each of the classes on the kymograph. The regions hand-drawn are the pixels of interest. The pixel values from the original kymograph are superimposed at these pixel locations in a separate image and the GWT features [28] or CT features, depending on the SVM classifier being trained, are computed at these pixel locations for each individual trail. Figure 3.8 and Figure 3.9 depict the extraction of training set for GWT features and CT features respectively. The training set consists of 100 examples for each of the 3 categories (i.e., 300 training examples in total).

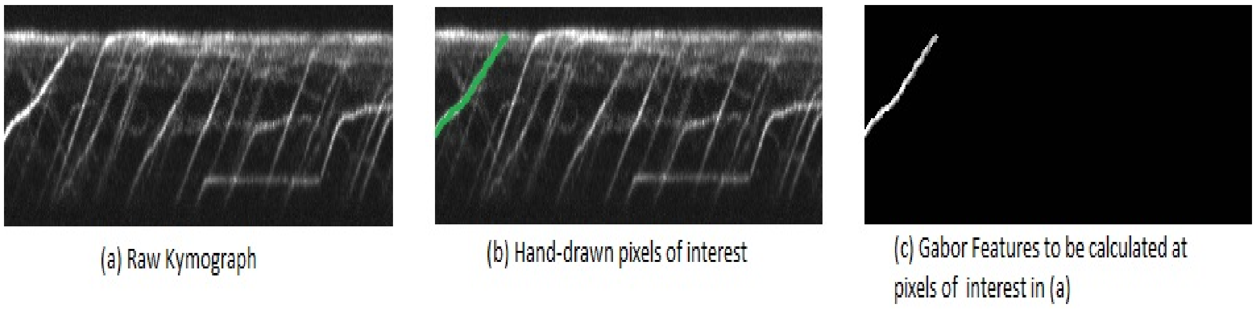
(a) Raw Kymograph
(b) Hand-drawn pixels of interest
(c) Gabor Features to be calculated at pixels of interest in (a)

Figure 3.8: Feature Image Creation-Gabor Features for anterograde trails

TESTING PROCEDURE

Segmentation of the test image is performed in the following manner for each of the two SVM classifiers:

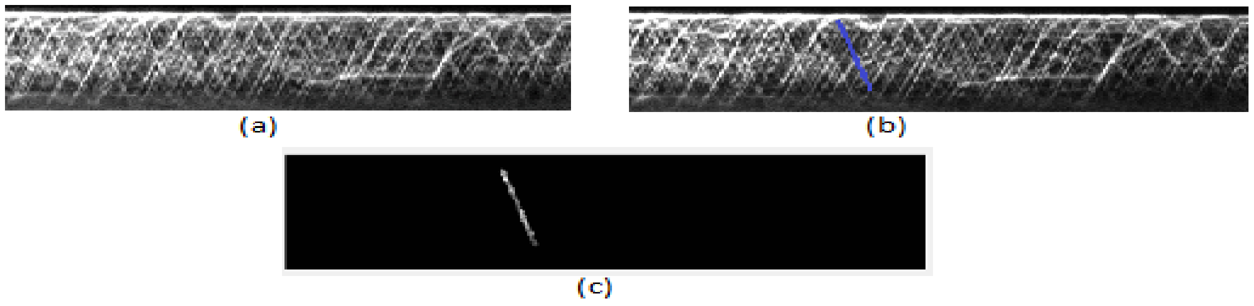

Figure 3.9: Feature Image Creation-Curvelet Features for retrograde trails

GWT feature-based SVM: The default GWT kernel is oriented at $\pi/2$. In order to extract edges in the specific directions needed for our purpose, the GWT kernel is oriented at $45°$ to extract anterograde trails and is convolved with the input kymograph image to obtain a resulting GWT image with edges in that direction. Similarly when the GWT kernel is oriented at $135°$ and convolved with the original image, it results in a GWT image with the retrograde trails. The angles of orientation of the GWT kernel are decided by computing the maximum likelihood of the angular orientation for each class from the training set.

CT feature-based SVM:

The Curvelet transform (CT) has been used in the past for the separation of bidirectional trails in a kymograph [29]. We have implemented a similar approach for the segmentation of a test image. We have implemented the CT scheme in MATLAB (The MathWorks) using CurveLab (http://www.curvelet.org) for the fast discrete CT. The steps of the scheme are discussed in greater detail.

The capability of the CT to extract directional edge features[30] at different orientations is exploited for segmenting edge features from a kymograph image. The CT provides details such as spectral information successfully at different orientations with reduced complexity. Curvelet decomposition using real coefficients and the wrapping function method is used because of its reduced complexity and faster computation. Curvelet transform decomposes the image in different direction and frequencies in the curvelet domain. As each CT coefficient c_{jlk} is associated with a particular location (the index k) and a particular direction (the index l), it is easy to use the CT coefficients to extract a field describing the directions and locations of major features in the image by the following procedure. We first select a number of CT levels $\{j_1, \ldots, j_P\}$, depending on the size of the image features we are interested in, typically the width of the edges. The selected levels are usually determined by trial and error, but it is generally better to include more than one level, as edges may vary in width and leave traces on several levels. Each selected level, ji, is associated with a grid $G_p = \{(k_1, k_2)|0 \leq k_1 < K_1^i, 0 \leq k_2 < K_2^i\}$ of size $K_1^i \times K_2^i$, determined by the discrete CT. Each curvelet coefficient c_{jlk} is associated with a direction determined by the index l. Since the number of directions varies with the curvelet level, with the number of directions doubling with every second level, the coefficients on coarser levels need to be mapped to all directions on the finest selected level that they overlap with. Now, for each direction l and location $k = (k_1, k_2)$ on the finest level j_p, we sum up the magnitudes of the curvelet coefficients. Having computed the magnitude, M_{lk}, we can now compute the major direction $l_0(k)$ at each grid point by:

$$l_0(k) = \arg\max_l M_{lk} \tag{3.5}$$

where $k \in G_p$ and define the field $\Psi(k) = (\Psi_1(k), \Psi_2(k))$, $k \in G_p$ as:

$$\Psi(k) = (M_{l_0k} \cdot \cos\theta_{l_0}, M_{l_0k} \cdot \sin\theta_{l_0}) \tag{3.6}$$

where θ_{l_0} is the angle associated with the direction l_0, by the definition of the discrete curvelet transform. The angle θ_{l_0} is taken along the valleys of the curvelets, meaning that the direction of the field $\Psi(k)$ will be along the edges in the image.

The directions chosen for the purpose of segmentation in the proposed system were $l = \{0, 1, 4, 5\}$ (i.e., capturing edges that occur in the first and third quadrants) for anterograde trails and $l = \{2, 3, 6, 7\}$ (i.e., capturing edges that occur in the second and fourth quadrants) for retrograde trails where $l \in \{0, 1, \ldots, 7\}$. The reason for selecting these quadrants is that all the anterograde trails in the training set occur in the first and third quadrants and all the retrograde trails lie in the second and fourth quadrants.

The post segmentation results after CT is applied to extract edges in different directions is shown in Figure 3.10

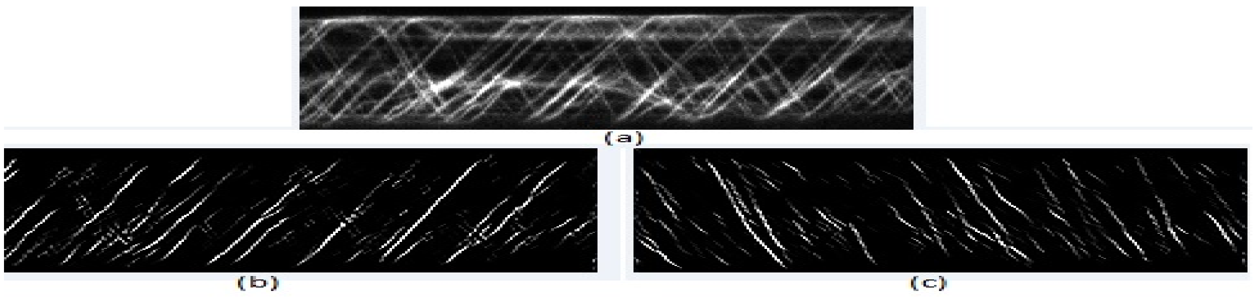

Figure 3.10: Post Segmentation Results using the Curvelet Transform (a) Raw Kymograph (b) Anterograde Trails (c) Retrograde Trails

Post Segmentation

After segmenting the IFT trails in the test image, each trail is fed to the respective SVM classifier and the classifier assigns it to one of the three aforementioned classes. All the IFT trails belonging to a certain class are mapped onto a separate image. Three images are generated after this step, one image for each trail class. Only the first two classes; anterograde and retrograde are of interest to us, so from here on we only deal with the two images containing the anterograde and retrograde trails. Next, the Sobel edge operator [31] is applied on these two images to enhance the trails. The Sobel operator masks are assigned values such that the gradients along the x and y directions are estimated properly, this ensures the enhancement of the trails in the required directions. The two images that are received as outputs of this process are then mapped onto the original kymograph for each of the classifiers (see Figure 4.1 & 4.2).

3.1.3 POST-PROCESSING

While comparing the post classification results, we see that the Gabor SVM detects a few more trails along the retrograde path, but they are staggered or detached whereas for Curvelet SVM detects fully attached trails. We perform some post processing on the Gabor SVM results to join the detached line segments.

The post-processing algorithm is used to join detached line segments along the same projected path and perform line fitting along each of the trails. After fitting line segments along each trail, the velocity of the trails are calculated. Figures 4.3and 4.4 show results after joining the staggering line segments and fitting a line segment along each trail. To join the staggering line fragments, we refer to the method used in [22] (Figure 4.3). First the two line segments are traced out independently and then they are lined up together if it is indicated that they belong to the same trajectory. Checking if two line segments belong to the same trajectory is based on relative orientation between two segments and relative distance between their end points. The orientation measure, θ , is defined as

$$\theta = \theta_a + \theta_b \tag{3.7}$$

where θ_a and θ_b are angles between the two line segments and the line joining the line segments and the connecting line. The relative distance between the two line segments is given by

$$d/(l_a + l_b) \tag{3.8}$$

where the lengths of the two line segments are denoted by l_a and l_b, snd d represents the distance between them. If $\theta < T_\sigma$ and $d/(l_a + l_b) > T_d$ where the threshold values T_σ and T_d

are 0.5 and 0.2, respectively, then l_a and l_b are connected. Figure 4.4 shows the final results after post processing.

3.1.4 VELOCITY-CALCULATION

We use polynomial regression, a form of linear regression to calculate the velocity of each detected trail. Linear regression[32] is the most basic and commonly used predictive analysis. Regression estimates are used to describe data and to explain the relationship between one dependent variable and one or more independent variables. At the center of the regression analysis is the task of fitting a single line through a scatter plot. Polynomial regression is a form of linear regression in which the relationship between the independent variable x and the dependent variable y is modelled as an nth degree polynomial 3.11. Polynomial regression is considered to be a special case of multiple linear regression. We implement this approach in matlab using the polyfit function.

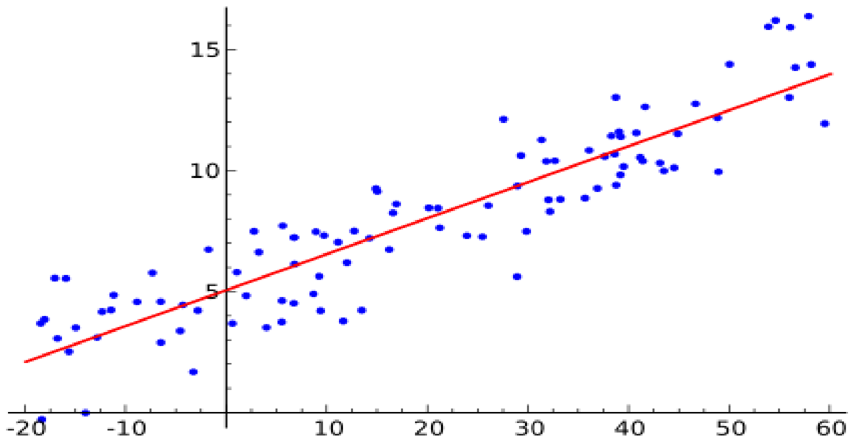

Figure 3.11: Regression analysis

Chapter 4

RESULTS

The main aim of our proposed methods is to reduce the time taken by biologists to manually analyze kymographs. A major aspect of this work is to train a classification model which takes in a test image and classifies the different trails into three different classes. The confusion matrices for GWT-based SVM classification and CT-based SVM classification are given in Tables 4.1 and 4.2. It is observed that the misclassification rate is very low especially for the retrograde and negative (stationary particles and background) IFT trails, in both cases. The misclassification rate is relatively higher for anterograde IFT trails in the case of GWT-SVM classification and is observed to mainly occur for very small trails which are mistaken as retrograde trails. However, the misclassification rate is still very low. Figures 4.5 and 4.6 show the velocity measure for each trail that is classified as anterograde or retrograde respectively. The average velocity over all anterograde trails is measured as 1.85µm/sec and average velocity over all retrograde trails is given by 3.37µm/sec when the trails are identified by the Gabor SVM. The average velocity over all anterograde trails is measured as 1.72µm/sec and average velocity over all retrograde trails is given by 3.29µm/sec when the trails are identified by the Curvelet SVM. When calculated manually, the average anterograde velocity is 1.75µm/sec and the average retrograde velocity is 3.13µm/sec, bringing the mean error to 0.03µm/sec and 0.16µm/sec for anterogrades and retrogrades respectively for the CT-SVM detected trails and 0.10µm and 0.20µm for anterogrades and retrogrades respectively for GWT- SVM detected trails.

We extend our experiments to calculate the average velocity for 10 kymographs using both approaches and notice that the error remains considerably less when compared with the manually calculated average velocity of the kymographs.Figures: 4.9, 4.10, 4.11 and 4.12.

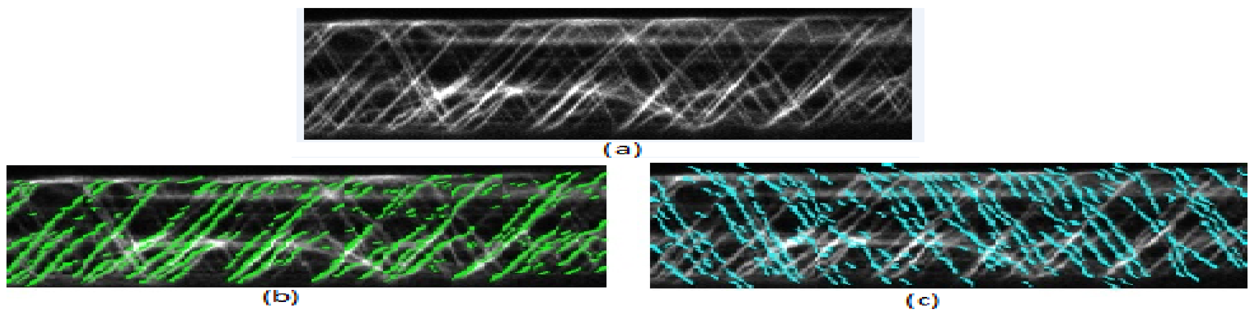

Figure 4.1: Post Classification Results Using the GWT (a) Raw Kymograph (b) Anterograde Trails (c) Retrograde Trails

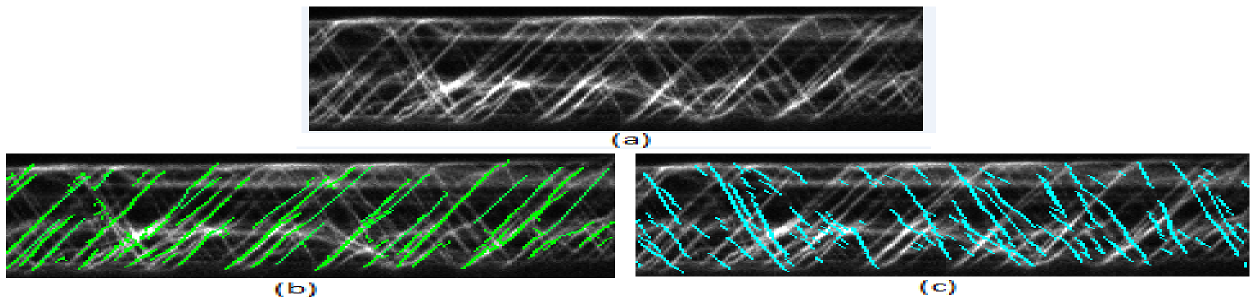

Figure 4.2: Post Classification Results Using the CT (a) Raw Kymograph (b) Anterograde Trails (c) Retrograde Trails

We compare our approaches with the approach discussed in [20]. In their approach, they train an LDA classifier to generate a probability map which shows the probability of each pixel belonging to a trail. They use Gabor filtered images and Frangi vesselness [33]

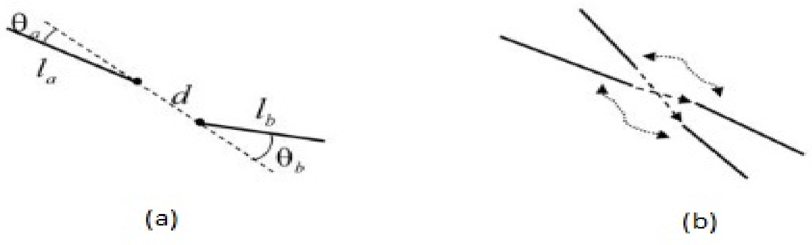

Figure 4.3: Post Classification Results (a) Orientation Measure (b) Distance Measure

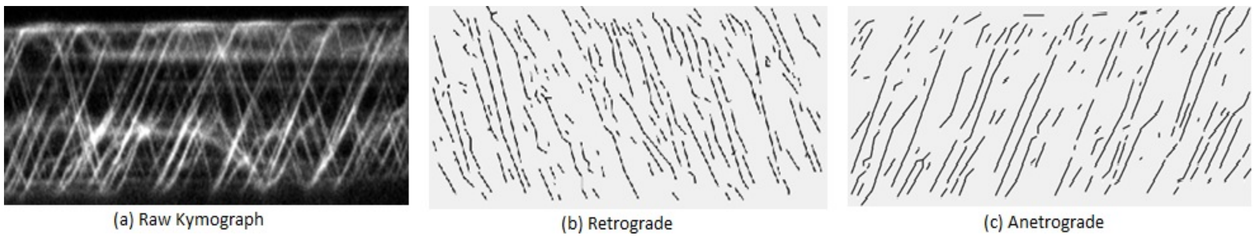

Figure 4.4: After Post-processing and Line-fitting is Performed on the Gabor SVM results (a) Raw Kymograph (b) Retrograde (c) Anterograde

measure to first enhance the trails. Then they hand-draw pixel values of interest from these images and label them as positive pixels and all other pixels are labeled as negative pixels. These are then used to train an LDA classifier and generate a probabilty map displaying the probability of each pixel belonging to a trail. We implemented their approach on our data set and observed that it did well for the anterograde IFT trails but not so well for the retrograde ones (Figure 4.13). Next, from Figure 4.13, we trace fragmented line segments along the same projected path for both the LDA generated probability map and the SVM classified results. The number of trails detected by each is shown in Table 4.3. If we consider the manual analysis as the baseline method, we see from the table that the SVM classifiers detect many more trails than the LDA classifier, which does not perform poorly with the anterograde trails but fails to detect most of the retrograde trails.

Table 4.1: Confusion Matrix After Classification: GWT-based SVM

Confusion Matrix After GWT-based SVM Classification			
Class	Anterograde	Retrograde	Negative
Anterograde	0.70	0.20	0.10
Retrograde	0.02	0.97	0.01
Negative	0.00	0.04	0.96

Table 4.2: Confusion Matrix After Classification: CT-based SVM

Confusion Matrix After Curvelet SVM Classification			
Class	Anterograde	Retrograde	Negative
Anterograde	0.86	0.04	0.10
Retrograde	0.12	0.82	0.06
Negative	0.02	0.00	0.98

Number of Trails Detected		
Method Used	Anterograde	Retrograde
Manual(Baseline)	32	31
LDA	25	12
SVM (GWT)	31	35
SVM (CT)	32	33

Table 4.3: Number of Trails Detected by Each Method

Average Localization Error σ		
Method Used	x-axis	y-axis
SVM (GWT): Anterograde	0.69	0.23
SVM (GWT): Retrograde	0.28	0.52
SVM (CT): Anterograde	0.45	0.21
SVM (CT): Retrograde	0.18	0.08

Table 4.4: Average Localization Error: In Terms of Number of Pixels

Figure 4.14 compares the manually detected lines with the SVM detected lines in both cases, i.e., the GWT-SVM and the CT-SVM. The green lines show the manually detected lines and the purple lines show the lines detected by each approach. It is observed that most lines are detected because of the pixel overlaps shown.The Gaussian filter applied to form the Gabor images smooths out the edge. We apply non-maxima suppression to the GWT-SVM classified image before comparing the classifier detected trails detected with the manually detected trails. Figure 4.14(a). Table4.4 shows average number of pixels by which the trails detected by each classifier are off from the manually traced trails along the x and y axes. It is a measure for the average localization error in terms of the number of pixels.

The time taken to identify the trails in a kymograph manually, varies depending on the clarity of the kymographs. For a kymograph with very clear trails it usually takes around 3-5 minutes for trail detection whereas for weaker kymographs it can take anywhere between 10-15 minutes to detect the trails of a single kymograph. The proposed algorithm takes on an average 98 seconds for the classification, post-processing and velocity estimation for each kymograph. Figure 4.18 shows the trails detected by our algorithm for a relatively weaker kymograph, that is when the trails cannot easily be distinguished from the noisy background by the naked human eye. Figure 4.18(b) shows the retrograde trails detected by the proposed algorithm. The lines are faint enough that they can easily be missed while trying to analyze the kymograph manually. Figure 4.19 shows the probability map generated when the LDA classifier from [20] is applied on the weak kymograph. It detects most of the anterograde IFT trails but once again fails to detect the retrograde ones.

We applied the proposed algorithm on the kymograph from [1] and the number of edges detected are significantly higher compared to the number of edges detected by the algorithm proposed in their paper as shown in Figure 4.15 and Figure 4.16. Figure 4.17 shows results to detect anterograde trails using Hough transform and our approach. The method proposed in [34] once again fails while trying to process noisy kymographs.

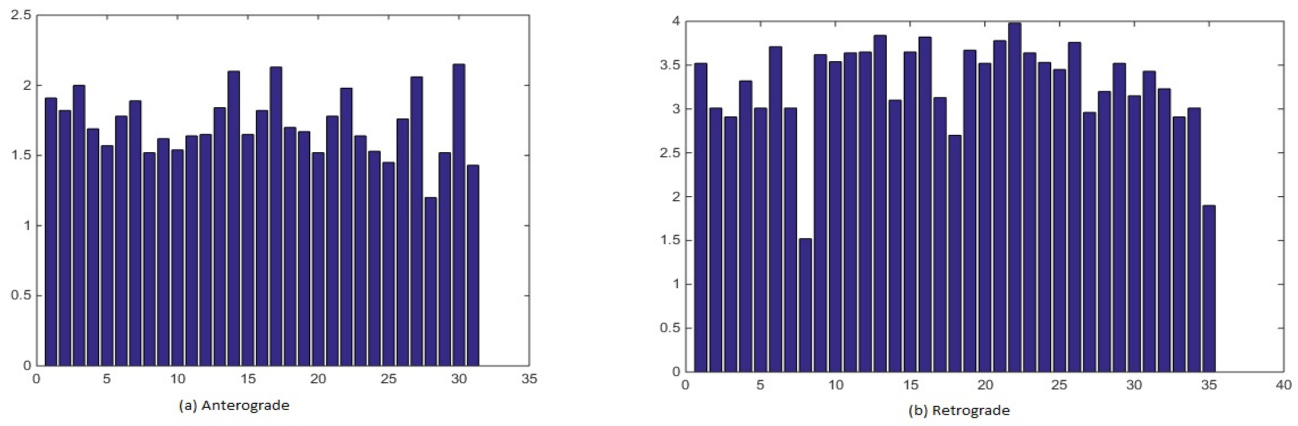

Figure 4.5: (a) Velocity for each identified anterograde trail(Gabor SVM) (b) Velocity for each identified retrograde trail(GaborSVM)

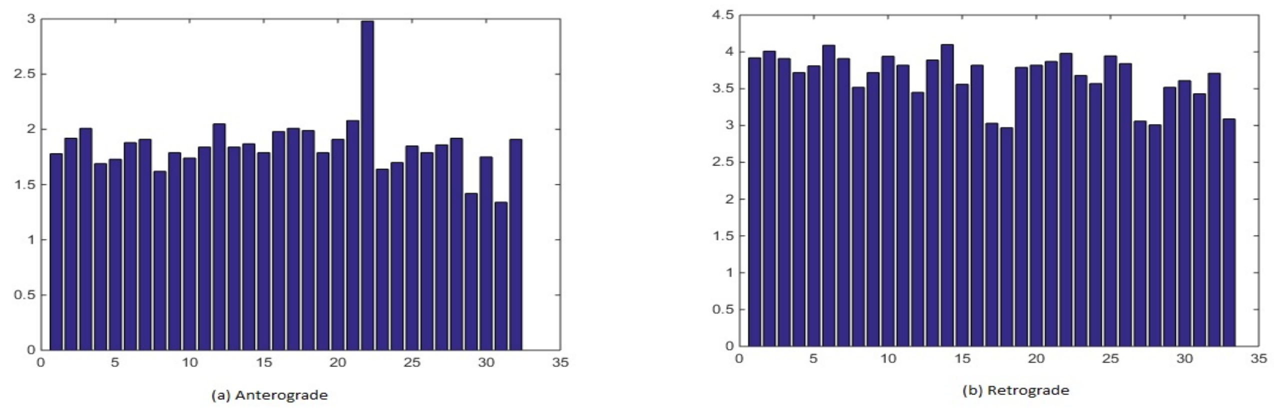

Figure 4.6: (a) Velocity for each identified anterograde trail(Curvelet SVM) (b) Velocity for each identified retrograde trail(Curvelet SVM)

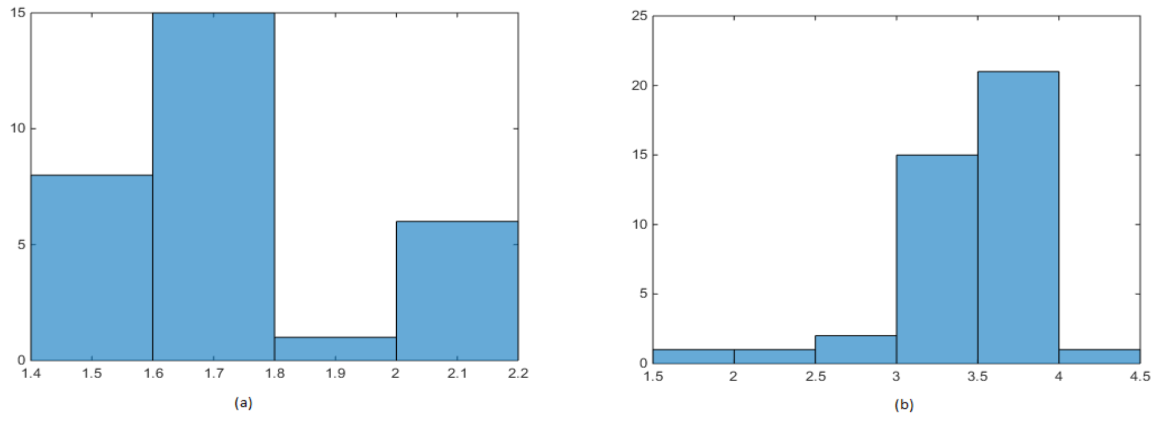

Figure 4.7: (a) Histogram showing velocity measure GWT-SVM detected trails: Anterograde (b) Histogram showing velocity measure GWT-SVM detected trails: Retrograde

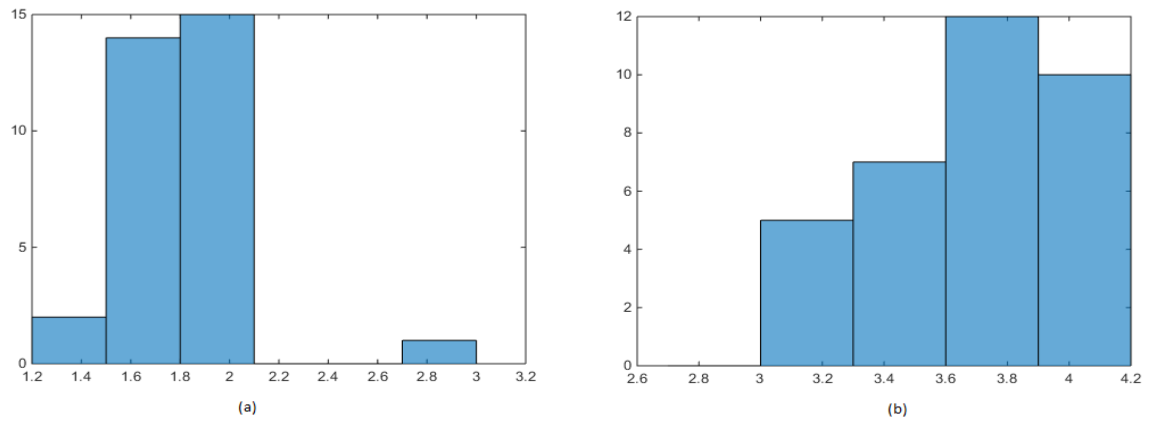

Figure 4.8: (a) Histogram showing velocity measure CT-SVM detected trails: Anterograde (b) Histogram showing velocity measure CT-SVM detected trails: Retrograde

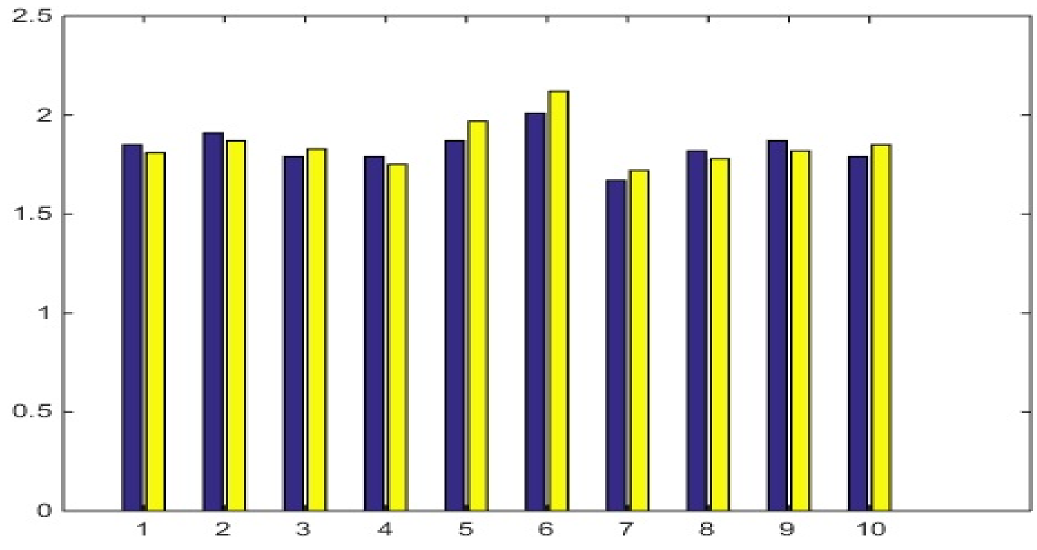

Figure 4.9: Average Anterograde Particle Velocity of Ten Kymographs(GWT-SVM) (a) Blue bar represents manually calculated velocity (b) Yellow bar represents manually calculated velocity

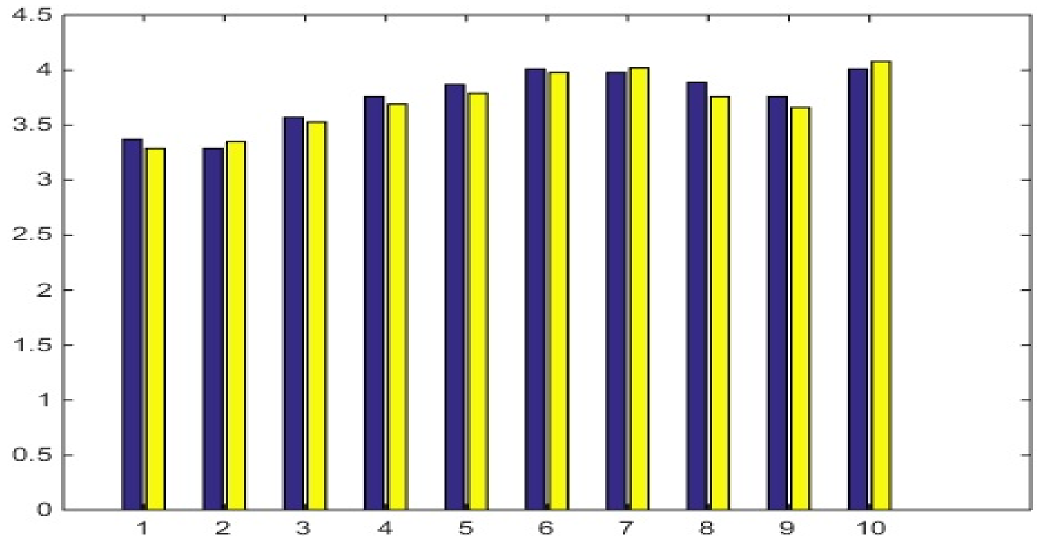

Figure 4.10: Average Anterograde Particle Velocity of Ten Kymographs(GWT-SVM) (a) Blue bar represents manually calculated velocity (b) Yellow bar represents manually calculated velocity

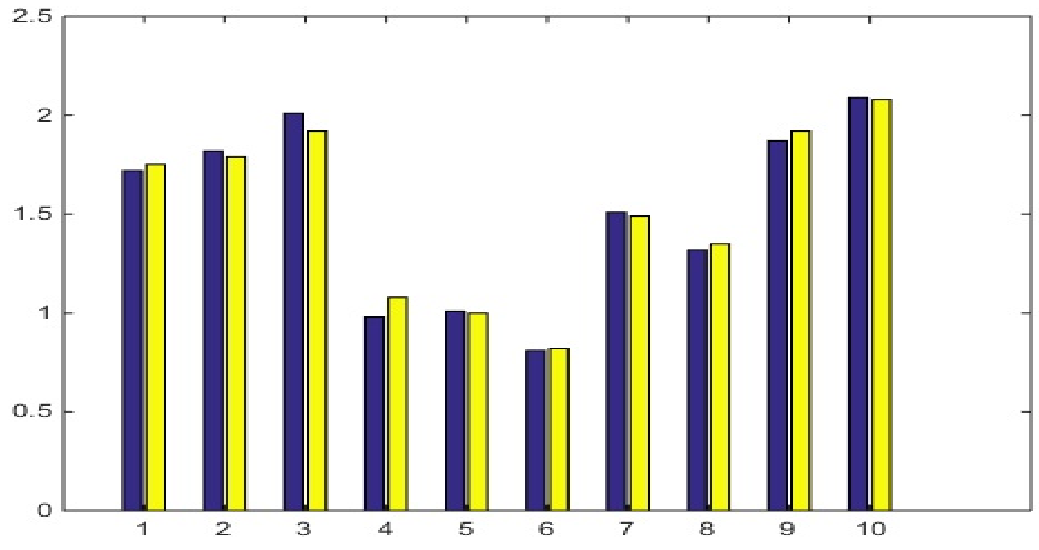

Figure 4.11: Average Anterograde Particle Velocity of Ten Kymographs(CT-SVM) (a) Blue bar represents manually calculated velocity (b) Yellow bar represents manually calculated velocity

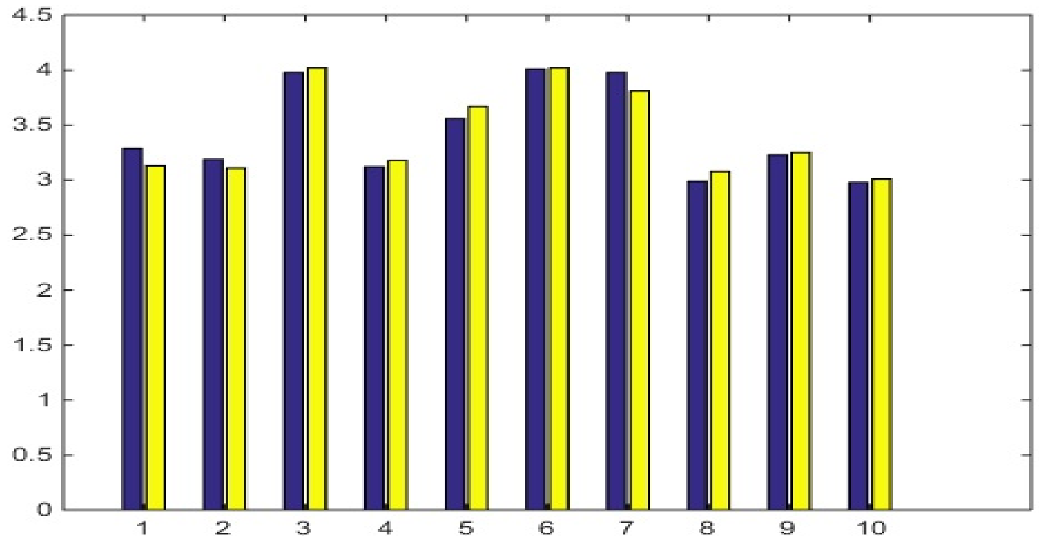

Figure 4.12: Average Retrograde Particle Velocity of Ten Kymographs(CT-SVM) (a) Blue bar represents manually calculated velocity (b) Yellow bar represents manually calculated velocity

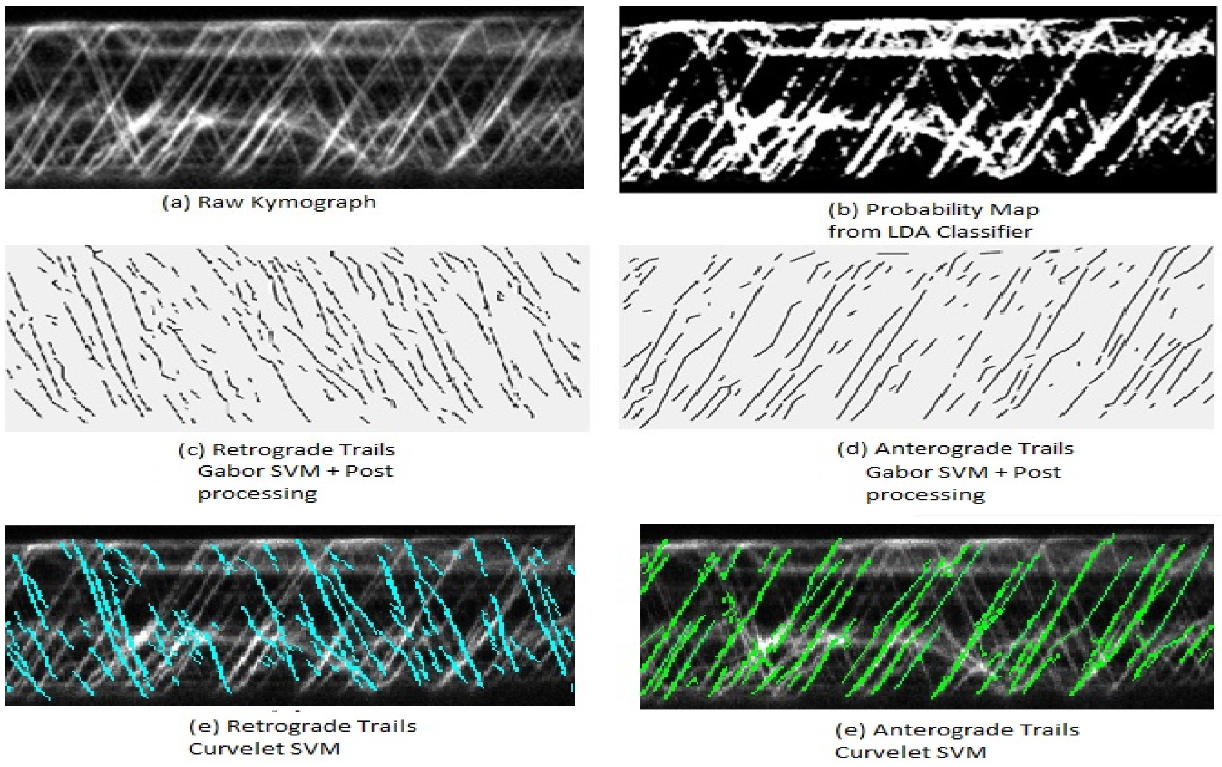
(a) Raw Kymograph
(b) Probability Map from LDA Classifier
(c) Retrograde Trails Gabor SVM + Post processing
(d) Anterograde Trails Gabor SVM + Post processing
(e) Retrograde Trails Curvelet SVM
(e) Anterograde Trails Curvelet SVM

Figure 4.13: (a) Raw Kymograph (b) Probability Map: LDA classifier (c) Retrograde Trails: Gabor SVM Classifier+ Post- processing(Proposed Method) (d) Anterograde Trails: Gabor SVM Classifier+ Post- processing(Proposed Method) (e) Retrograde Trails: Curvelet SVM Classifier(Proposed Method) (f) Anterograde Trails: Curvelet SVM Classifier(Proposed Method)

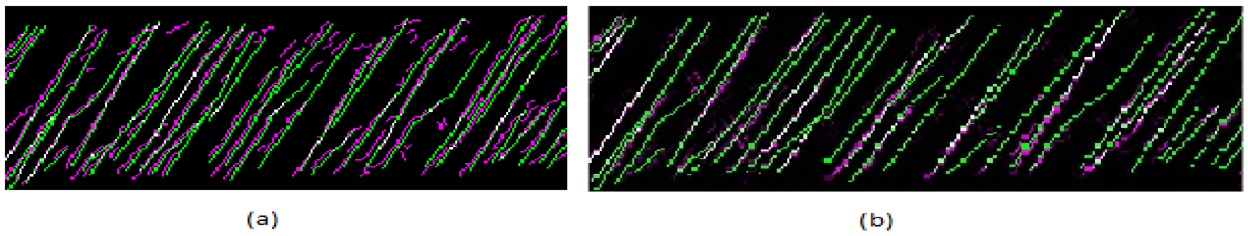

Figure 4.14: (a) Gabor SVM (b) Curvelet SVM

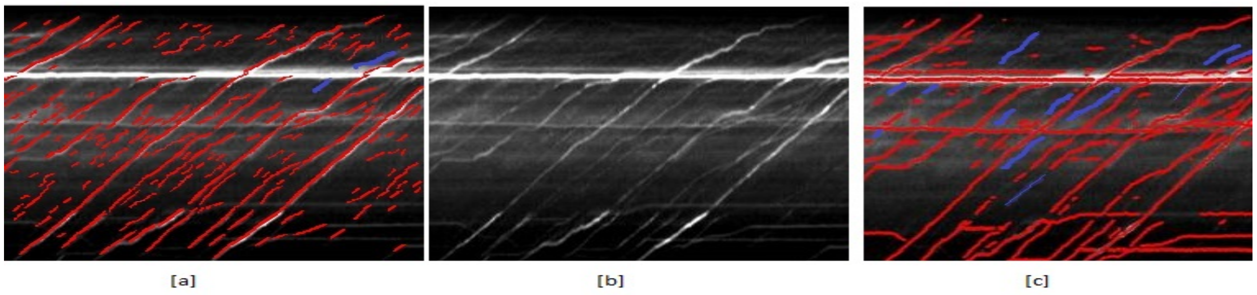

Figure 4.15: [b] Raw kymograph from [1] [a] shows result of our proposed method: Gabor SVM on [1] [c] shows result of [b]. The red lines depict the lines detected by the algorithms and the blue lines show the missed out lines.

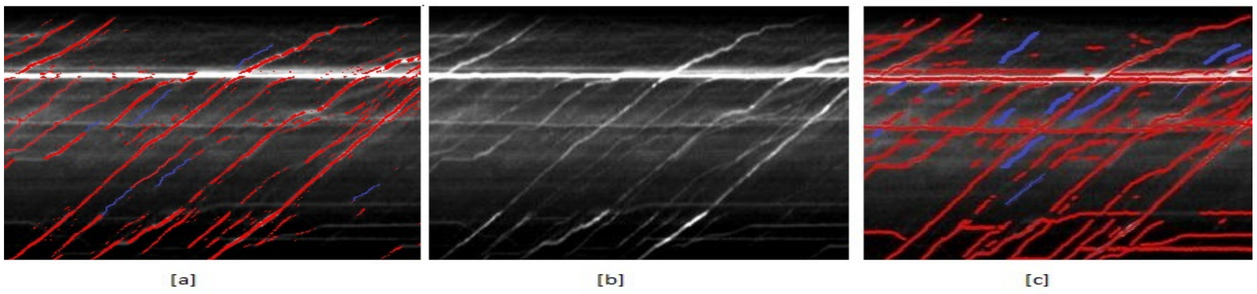

Figure 4.16: [b] Raw kymograph from [1] [a] shows result of our proposed method: Curvelet SVM on [1] [c] shows result of [b]. The red lines depict the lines detected by the algorithms and the blue lines show the missed out lines.

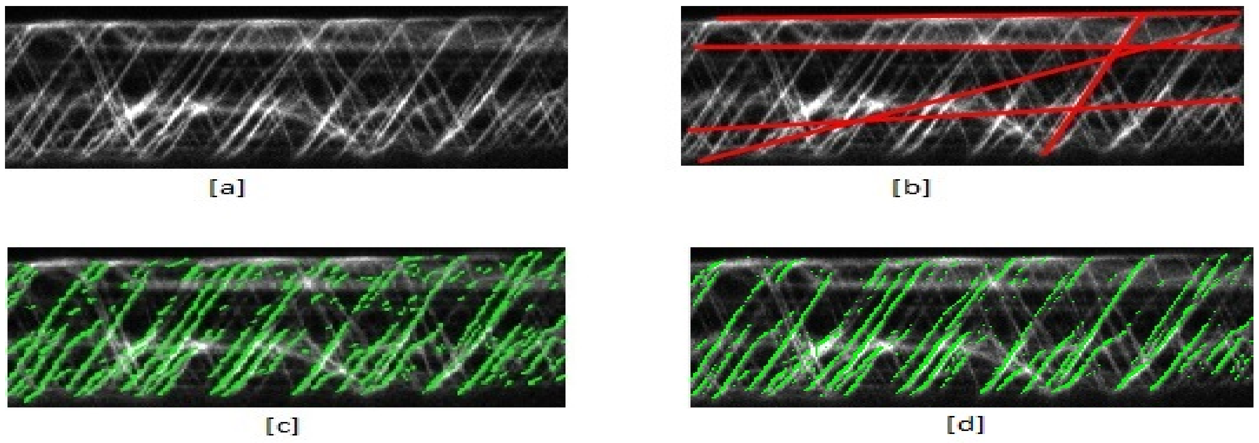

Figure 4.17: [a] Raw kymograph [b]Hough tranform to detect anterograde trails. Red lines are the trails detected [c]. The green lines depict the anterograde lines detected by our algorithm: GWT-based SVM before post processing.

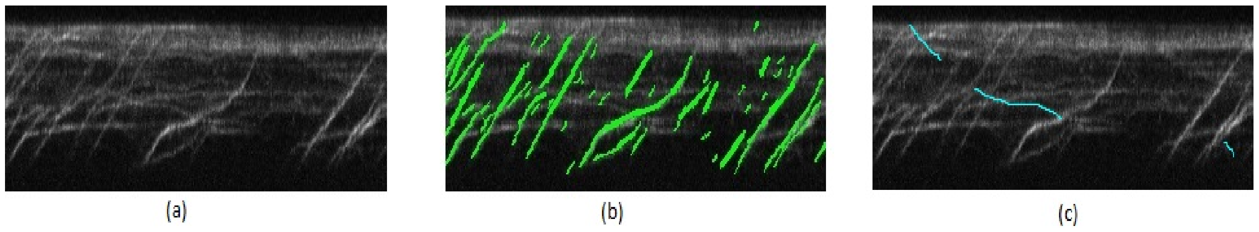

Figure 4.18: (a) Weak Kymograph(Raw) (b) Anterograde Trails: Gabor SVM Classifier(Proposed Method) (c) Retrograde Trails: Gabor SVM Classifier(Proposed Method)

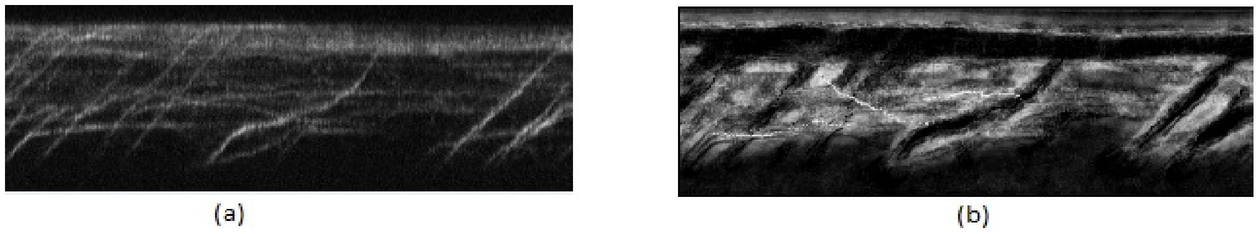

Figure 4.19: (a) Weak Kymograph(Raw) (b) Probability Map: LDA classifier

Chapter 5

SUMMARY AND CONCLUSIONS

The main aim of our proposed approach is to automate the analysis of kymographs and document movements and transport as they occur in axons, in cilia, and other cell extensions (filopodia etc.). The immediate goal is extraction of particle velocities from large data sets in a reproducible manner and calculating the frequencies of anterograde and retrograde particles. Our algorithm involves training two SVM classifiers to classify the different types of trails, segment a test image to classify the different trails in three different groups and calculate the velocity of these trails. Our algorithm works well for kymographs with very clear trajectories and also for kymographs where the trails cannot be easily detected. It reduces human effort and gives continuous instantaneous velocities even when there are multiple anterograde and retrograde trails which cross each other, hence increasing the complexity. Our future work will include calculating parameters other than the velocity of the trails, like the run length, pause time and pause frequency. There are also instances where two trails merge, this is when the two molecules combined to form one trail. Calculating the diffused velocity of the merged molecule from this trail is also going to be a significant contribution to our future work.

Bibliography

[1] K. Zhang, Y. Osakada, W. Xie, and B. Cui, "Automated image analysis for tracking cargo transport in axons," *Microscopy Research and Techniques*, vol. 74, no. 7, pp. 605–613, 2011.

[2] L. T. Haimo and J. L. Rosenbaum, "Cilia, flagella, and microtubules." *The Journal of Cell Biology*, vol. 91, no. 3, pp. 125s–130s, 1981.

[3] P. Satir and S. T. Christensen, "Structure and function of mammalian cilia," *Histochemistry and Cell Biology*, vol. 129, no. 6, pp. 687–693, 2008.

[4] K. F. Jarrell, *Pili and Flagella: Current Research and Future Trends.* Horizon Scientific Press, 2009.

[5] D. G. Cole, D. R. Diener, A. L. Himelblau, P. L. Beech, J. C. Fuster, and J. L. Rosenbaum, "Chlamydomonas kinesin-ii–dependent intraflagellar transport (ift): Ift particles contain proteins required for ciliary assembly in caenorhabditis elegans sensory neurons," *The Journal of Cell Biology*, vol. 141, no. 4, pp. 993–1008, 1998.

[6] C. A. Wagner, "News from the cyst: insights into polycystic kidney disease," *Journal of Nephrology*, vol. 21, no. 1, p. 14, 2008.

[7] W. R. Cnossen and J. P. Drenth, "Polycystic liver disease: an overview of pathogenesis, clinical manifestations and management," *Orphanet Journal of Rare Diseases*, vol. 9, no. 1, p. 1, 2014.

[8] M. Brueckner, "Heterotaxia, congenital heart disease, and primary ciliary dyskinesia," *Circulation*, vol. 115, no. 22, pp. 2793–2795, 2007.

[9] M. Adams, U. M. Smith, C. V. Logan, and C. A. Johnson, "Recent advances in the molecular pathology, cell biology and genetics of ciliopathies," *Journal of Medical Genetics*, vol. 45, no. 5, pp. 257–267, 2008.

[10] K. F. Lechtreck, "In vivo imaging of ift in chlamydomonas flagella," *Methods Enzymol*, vol. 524, pp. 265–284, 2013.

[11] R. Y. Tsien, "The green fluorescent protein," *Annual Review of Biochemistry*, vol. 67, no. 1, pp. 509–544, 1998.

[12] N. C. Shaner, G. G. Lambert, A. Chammas, Y. Ni, P. J. Cranfill, M. A. Baird, B. R. Sell, J. R. Allen, R. N. Day, M. Israelsson *et al.*, "A bright monomeric green fluorescent protein derived from branchiostoma lanceolatum," *Nature Methods*, vol. 10, no. 5, pp. 407–409, 2013.

[13] J. Buisson, N. Chenouard, T. Lagache, T. Blisnick, J.-C. Olivo-Marin, and P. Bastin, "Intraflagellar transport proteins cycle between the flagellum and its base," *Journal of Cell Science*, vol. 126, no. 1, pp. 327–338, 2013.

[14] T. S. Lee, "Image representation using 2d gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[15] J. Ma and G. Plonka, "The curvelet transform," *IEEE Signal Processing Magazine*, vol. 27, no. 2, pp. 118–133, 2010.

[16] S. Andrews, J. Gilley, and M. P. Coleman, “Difference tracker: Imagej plugins for fully automated analysis of multiple axonal transport parameters,” *Journal of Neuroscience Methods*, vol. 193, no. 2, pp. 281–287, 2010.

[17] W. B. Ludington and W. F. Marshall, “Automated analysis of intracellular motion using kymographs in 1, 2, and 3 dimensions,” in *SPIE BiOS: Biomedical Optics.* International Society for Optics and Photonics, 2009, pp. 71 840Y–71 840Y.

[18] D. H. Ballard, “Generalizing the hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[19] O. Welzel, J. Knörr, A. M. Stroebel, J. Kornhuber, and T. W. Groemer, “A fast and robust method for automated analysis of axonal transport,” *European Biophysics Journal*, vol. 40, no. 9, pp. 1061–1069, 2011.

[20] A. Nair, S. Ramanarayanan, S. Ahlawat, S. Koushika, N. Joshi, and M. Sivaprakasam, “Axonal transport velocity estimation from kymographs based on curvilinear feature extraction and spline fitting,” in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* IEEE, 2014, pp. 4240–4243.

[21] P. Mangeol, B. Prevo, and E. J. Peterman, “Kymographclear and kymographdirect: two tools for the automated quantitative analysis of molecular and cellular dynamics using kymographs,” *Molecular Biology of the Cell*, pp. mbc–E15, 2016.

[22] A. Mukherjee, B. Jenkins, C. Fang, R. J. Radke, G. Banker, and B. Roysam, “Automated kymograph analysis for profiling axonal transport of secretory granules,” *Medical Image Analysis*, vol. 15, no. 3, pp. 354–367, 2011.

[23] M. Qiu, H.-C. Lee, and G. Yang, “Nanometer resolution tracking and modeling of bidirectional axonal cargo transport,” in *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI).* IEEE, 2012, pp. 992–995.

[24] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer Science & Business Media, 2013.

[25] F. Jiao, W. Gao, L. Duan, and G. Cui, "Detecting adult image using multiple features," in *Info-tech and Info-net, 2001. Proceedings. ICII 2001-Beijing. 2001 International Conferences on*, vol. 3. IEEE, 2001, pp. 378–383.

[26] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory.* ACM, 1992, pp. 144–152.

[27] T. Joachims, "Making large scale svm learning practical," Universität Dortmund, Tech. Rep., 1999.

[28] R. M. Rangayyan, F. J. Ayres, F. Oloumi, F. Oloumi, and P. Eshghzadeh-Zanjani, "Detection of blood vessels in the retina with multiscale gabor filters," *Journal of Electronic Imaging*, vol. 17, no. 2, pp. 023 018–023 018, 2008.

[29] N. Chenouard, J. Buisson, I. Bloch, P. Bastin, and J.-C. Olivo-Marin, "Curvelet analysis of kymograph for tracking bi-directional particles in fluorescence microscopy images," in *2010 IEEE International Conference on Image Processing.* IEEE, 2010, pp. 3657–3660.

[30] T. Gebäck and P. Koumoutsakos, "Edge detection in microscopy images using curvelets," *BMC bioinformatics*, vol. 10, no. 1, p. 1, 2009.

[31] M. Juneja and P. S. Sandhu, "Performance evaluation of edge detection techniques for images in spatial domain," *International Journal of Computer Theory and Engineering*, vol. 1, no. 5, p. 614, 2009.

[32] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied linear statistical models.* Irwin Chicago, 1996, vol. 4.

[33] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 1998, pp. 130–137.

[34] O. Welzel, D. Boening, A. Stroebel, U. Reulbach, J. Klingauf, J. Kornhuber, and T. W. Groemer, “Determination of axonal transport velocities via image cross-and autocorrelation,” *European Biophysics Journal*, vol. 38, no. 7, pp. 883–889, 2009.