8-2020

Topological Data Analysis in Sub-cellular Motion Reconstruction and Filament Networks Classification

Le Yin
University of Tennessee, lyin6@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Le Yin entitled "Topological Data Analysis in Sub-cellular Motion Reconstruction and Filament Networks Classification." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Mathematics.

Vasileios Maroulas, Major Professor

We have read this dissertation and recommend its acceptance:

Christopher Strickland, Andreas Nebenführ, Steven Abel, Haileab Hilafu

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Topological Data Analysis in Sub-cellular Motion Reconstruction and Filament Networks Classification

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Le Yin

August 2020

This dissertation is dedicated to my beloved parents for their endless love and support.

Acknowledgments

I would like to express my deepest appreciation and thankfulness to my advisor Dr. Vasileios Maroulas for his endless support and patient guidance. I could not finish this work without his careful help and encouragement. I would like to thank my advisory committee members Dr. Christopher Strickland, Dr. Andreas Nebenführ, Dr. Steven Abel, and Dr. Haileab Hilafu for their valuable discussion and suggestions. I am honored to have Dr. Ioannis Sgouralis and Ephy Love as my collaborators. Their expertise and perspectives have greatly enriched my educational and research experience. I gratefully acknowledge the NSF Grant MCB-1715794 for supporting my work.

I also want to thank my colleagues Honor Akenuwa, Dr. Farzana Nasrin, Adam Spannaus, Cassie Micucci and Christopher Oballe in our reseach group for their endless help. I would like to thank my friends too, especially Bo Gao, Xiaoyan Pan, Wenqiang Feng, Delong Li and James Ren. Thank you for your encouragement and support during my studies and life. Finally, I would like to add a special thanks to my dear parents for their endless love, support and encouragement.

Abstract

Topological Data Analysis is a powerful tool in the image data analysis. In this dissertation, we focus on studying cell physiology by the sub-cellular motions of organelles and generation process of filament networks, relying on topology of the cellular image data.

We first develop a novel, automated algorithm, which tracks organelle movements and reconstructs their trajectories on stacks of microscopy image data. Our tracking method proceeds with three steps: (i) identification, (ii) localization, and (iii) linking, and does not assume a specific motion model. This method combines topological data analysis principles with Ensemble Kalman Filtering in the computation of associated nerve during the linking step. Moreover, we show a great success of our method with several applications.

We then study filament networks as a classification problem, and propose a distance-based classifier. This algorithm combines topological data analysis with a supervised machine learning framework, and is built based on the foundation of persistence diagrams on the data. We adopt a new metric, the d_p^c distance, on the space of persistence diagrams, and show it is useful in catching the geometric difference of filament networks. Furthermore, our classifier succeeds in classifying filament networks with high accuracy rate.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Cell physiology depends on the motion of sub-cellular structures. Cytosolic streaming and organelle motility are critical factors [8, 10, 19, 64, 84]. Such intracellular motion is particularly pronounced in plant cells and is known to be essential to many cellular functions including growth and overall health [55]. In particular, organelle motility in plant cells is driven by motor proteins that directionally move along myosin filaments or diffuse in the cell sap and occasionally switch between these modes. Additionally, different motor proteins generate patterns of motion with different characteristics such as speed, turning angles, switching frequencies between different motions. Due to the complex nature of these underlying dynamics, an understanding of organelle motility based on first principles remains uncharacterized. Instead, intracellular motion is commonly studied experimentally.

Cell physiology also depends on actin cytoskeleton and actin filament. The actin cytoskeleton is a complex network of proteins that is present in all eukaryotic cells. In addition to its function as cellular scaffolding, the actin cytoskeleton enables several basic cellular functions including the control of cellular shape and direction of movement [86]. These basic functions are critical to many higher order physiological processes such as cell division, expansion, mobility and motility[22].

In the actin cytoskeleton, actin filament organization is thought to be partially governed by the interaction of filaments and partially by myosin motor proteins. Actin filaments are polar structures, ploymerized by globular actin proteins. Many actin-binding proteins have potential to bind to actin filaments at various sites along the filament. These proteins allow

actin filaments to assemble and disassemble spatiotemporally, and cross-link into networks at multiple binding sites. To understand certain behaviors of cells, it is tremendous important to understand the processes that govern the generation of actin filament networks. A key driver of these processes may be the relationship between actin-binding proteins, individual filaments and emergent networks.

One way to study the cell physiology is by intracellular organelle motion. Conventional fluorescence microscopy is one of the most popular technique employed for the direct observation of intracellular motion [61, 59, 62, 88]. With the well-engineered optical equipment and the fast development of bio-molecular labeling techniques, it is now routine to observe organelle dynamics. This in turn has led to the acquisition of vast datasets. A thorough and accurate reconstruction of organelle trajectories in these datasets is a necessary task to distinguish motor protein structures, elucidate their behavior, and globally characterize their motility. To accomplish this, some studies analyze raw measurements and track sub-cellular motions manually which has yielded estimates with good accuracy [17, 25, 26, 48, 61]. Nevertheless, manual tracking is tedious, time consuming, unreproducible and unrealistic for complex datasets with multiple simultaneous motions, especially those encountered in plant microscopy.

Automated tracking algorithms, capable of analyzing organelle motility datasets, provide an opportunity to overcome these difficulties and robustly track a large number of sub-cellular targets. These automated processes offer tighter error bounds and the promise of overcoming the low throughput of manual tracking. Additionally, automated tracking algorithms can reveal large scale motion patterns during the entire time course of an imaging experiment [18]. Therefore, developing automated intercellular tracking algorithms for organelles, specifically designed for plant cell imaging, are essential.

Intracellular tracking can be broken down into four steps: (i) identification, where the number of moving organelles is estimated first; (ii) localization, where the position of each identified organelle is detected in space throughout time; (iii) linking, where estimated localizations belonging to the same organelle trajectory are connected over time; and (iv) interpretation, where the estimated trajectories are used to derive quantitative information about the organelle motion [42]. Many methods for multiple targets tracking have been

developed so far [3, 6, 16, 33, 39, 37, 42, 55, 56, 67, 69, 79, 80, 85], but only few of them focus specifically on microscopy image data, while others are not applicable to image data or fail due to the introduction of missassignments. The study in [55] provides a solution by using a Bayesian framework that considers the intracellular movements as members of a set and set tracking is the tracking place, [33] developes a tracking approach that combine tracking information in the optimization procedure, [16] presents a method for simultaneously tracking thousands of targets by adapting the multiple hypothesis tracking algorithm, [75] solves the problem by considering a topological linking technique with minimal assumptions about the underlying dynamics. In [42], a survey of all techniques applicable to image data is provided.

In this dissertation, we seek to improve the linking stage as the first half of our work. We propose an automated algorithm based on Bayesian identification of organelle parameters, Ensemble Kalman filter (EnKF) estimations of displacement fields and topological linking on the trajectories space. A Bayesian framework is applied to identify important parameters of organelles. Subsequently, we use EnKF to estimate the displacements of organelles since our linking method is based on these displacements. The linking process is completed by using a topological data analysis (TDA) technique [12, 20, 75, 77] to find the geometry of the data space. This embeds the data into a topological space, in which the trajectories are reconstructed by identifying connected components.

Next we focus on the actin filament network generating process, which can be artificially realized by simulating the dynamic structure of filament networks, combining theory from theoretical physical with experimental stochastic simulation. With this technology one controls the known factors, which will affect the structure of networks. Varying these initial conditions enables researchers to compare the conditional difference in outcomes of the simulated networks. This experimental strategy can provide an opportunity to independently examine the role each factor plays in the process. These factors could include the cross-linker density(number of cross-linkers per certain area), cross-linker stiffness, maximum angle that exist between two filament segments to be crosslinked, and so on[22, 23]. In this dissertation, we propose a machine learning approach to classify filament networks generated by a varied

cross-linker density using TDA technique. Our method leverages the topology of the actin networks through TDA.

Precisely, our classifier uses persistent homology to measure differences in topological features. Persistent homology records the appearance and disappearance of homological features, the connected components and holes in the filament network data. We encode the homological features of filaments networks into persistence diagrams and classify simulated actin networks by calculating the similarities in their persistence space. Our exploratory work is the first time filament networks have been studied as a classification problem. This work could serve as a pilot for future research in actin cytoskeleton organization. In the future, this work should be useful in the course of research on cytoplasmic streaming to classify real cells based on images of their actin networks. This would provide biologists a method of disentangling the interaction of myosin motor proteins, the actin network, and streaming, i.e., by imaging the actin structure and clustering cells based on their actin network topology, the researcher may be able to fix a network structure while varying parameters specific to myosin.

The rest of this dissertation is organized as follows. In Chapter 2, background knowledge of filtering algorithms and TDA are provided, we focus on introduction of Kalman filter, Ensemble Kalman filter, simplicial complexes and persistence diagrams. In Chapter 3, we formulate the problem, give the technical details of the automated intercellular tracking algorithm, and show the results when our method is applied to simulated and real data sets. In Chapter 4, we describe the filament network data, demonstrate one distance-based algorithm for classifying filament networks and exhibit the numerical results. Finally, conclusion and discussion are presented in Chapter 5.

Chapter 2

Background

2.1 Background of filtering algorithms

Filtering algorithms are wildly used in tracking problems. Two of the most representative filtering algorithms are Kalman filter and particle filter. In this section, we will introduce principle and formulation of the Kalman filter (KF) and Ensemble Kalman filter (EnKF).

2.1.1 Kalman filter

Kalman filter has solved the problem of estimating the state in a discrete time process system with noisy sensor measurements [5]. The process is normally controlled by a linear Gaussian stochastic equation

$$x_k = Ax_{k-1} + u_{k-1},\ u_{k-1} \sim N(0, Q), \tag{2.1}$$

with a measurement

$$z_k = Hx_k + v_k,\ v_k \sim N(0, R). \tag{2.2}$$

In Eq. (2.1), $\{x_k\}$ are states in $\mathbb{R}^n$, A is a $n \times n$ matrix, $\{u_{k-1}\}$ are process running noises, which are independent and identically distributed and follow a multivariate normal distribution with mean 0 and covariance Q. This equation relates state x_k at step k to its

previous state at step $k-1$. In Eq. (2.2), z_k is the measurement of x_k in $\mathbb{R}^m$, H is $m \times n$ matrix, $\{v_k\}$ are measurement noises, which are also independent and identically distributed and follow a multivariate normal distribution with mean 0 and covariance R. This equation gives the relation between the state and measurement at the same step k.

Define $\hat{x}'_k$ as an *a priori* estimate of state at step k. It is an estimation based on knowledge of the process before step k. Define $\hat{x}_k$ as an *a posteriori* estimate of state after the measurement z_k at step k is obtained. The main idea of Kalman filter is that we want to correct an *a priori* estimation with the feedback from the measurement to get an *a posteriori* estimation, i.e., our goal is to write an *a posteriori* estimation as a weighted sum of the *a priori* estimation and difference between measurement and predicted measurement,

$$\begin{aligned} \hat{x}_k &= \hat{x}'_k + K_k(z_k - \hat{z}_k) \\ &= \hat{x}'_k + K_k(z_k - H\hat{x}'_k). \end{aligned} \tag{2.3}$$

In the equation above, $z_k - H\hat{x}'_k$ is called *innovation* or *residual*. The innovation equals 0 indicates the agreement of measurement and predicted measurement. K_k is called *Kalman Gain*. It is calculated by minimizing the *a posteriori* estimate error covariance. We are going to show the detailed steps of finding K_k in the following paragraph.

The *a posteriori* estimate error covariance and the *a priori* estimate error covarican are written as

$$\begin{aligned} P_k &= E\left[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\right], \\ P'_k &= E\left[(x_k - \hat{x}'_k)(x_k - \hat{x}'_k)^T\right], \end{aligned} \tag{2.4}$$

repectively. Then substitute Eqs. (2.3)(2.2) into Eq. (2.4), get

$$\begin{aligned} P_k &= E\left[\left[(I - K_kH)(x_k - \hat{x}'_k) - K_kv_k\right]\left[(I - K_kH)(x_k - \hat{x}'_k) - K_kv_k\right]^T\right] \\ &= (I - K_kH)E\left[(x_k - \hat{x}'_k)(x_k - \hat{x}'_k)^T\right](I - K_kH)^T + K_kE[v_kv_k^T]K_k^T \\ &= (I - K_kH)P'_k(I - K_kH)^T + K_kRK_k^T \\ &= P'_k - K_kHP'_k - P'_kH^Tk_k^T + K_k\left(HP'_kH^T + R\right)K_k^T. \end{aligned} \tag{2.5}$$

Take trace of both sides in Eq. (2.5)

$$[P_k] = [P'_k] - [K_k H P'_k] - \left[P'_k H^T k_k^T\right] + \left[K_k \left(H P'_k H^T + R\right) K_k^T\right],$$

and further take derivative respect to K_k

$$\frac{d\,[P_k]}{dK_k} = -2(HP'_k)^T + 2K_k \left(H P'_k H^T + R\right).$$

Set the derivative result equals 0 and sovle for K_k, we get the Kalman Gain as

$$K_k = P'_k H^T (H P'_k H^T + R)^{-1}. \tag{2.6}$$

Then substitute Eq. (2.6) back into Eq. (2.5), P_k can be further simplified as

$$\begin{aligned} P_k &= P'_k - P'_k H^T (H P'_k H^T + R)^{-1} H P'_k \\ &= P'_k - K_k H P'_k \\ &= (I - K_k H) P'_k. \end{aligned}$$

Overall, Kalman filter can be summarized into five equations in Algorithm 1. The first two equations are predicting equations. In these two equations, an *a priori* estimate is made and the process noise covariance is updated. The next three equations are updating equations. In these three equations, the Kalman gain is calculated, a *a posteriori* estimate is computed as a combination of the *a priori* estimate and weighted measurement innovation, and then the process noise covariance is further updated. Note that, as $\lim_{R\to 0} K_k = H^{-1}$ and $\lim_{R\to 0} \hat{x}_k = z_k$, thus when the measurement error covariance R approaches 0, the Kalman gain K_k weights the innovation more heavily; conversely, since $\lim_{P'_k\to 0} K_k = 0$ and $\lim_{P'_k\to 0} \hat{x}_k = \hat{x}'_k$, therefore when the *a priori* estimate error covariance P'_k approaches 0, the Kalman gain K_k weights the *a priori* estimate more heavily instead [5]. In general, the expected *a posteriori* error is kept minimized in a long term run.

Algorithm 1 Kalman filter

$$
\begin{aligned}
\hat{x}'_k &= A\hat{x}_{k-1} \\
P'_k &= AP_{k-1}A^T + Q \\
K_k &= P'_k H^T (HP'_k H^T + R)^{-1} \\
\hat{x}_k &= \hat{x}'_k + K_k(z_k - H\hat{x}'_k). \\
P_k &= (I - K_k H)P'_k
\end{aligned}
$$

2.1.2 Ensemble Kalman filter

Kalman filter only works for the case when the process is governed by a linear Gaussian stochastic equation. But in the real world problems, we are facing nonlinear processes most of the time, i.e.

$$x_k = \Psi(x_{k-1}) + u_{k-1}, \;\; u_{k-1} \sim N(0, Q), \tag{2.7}$$

where $\Psi : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a nonlinear function. Thus a more robust and universal method is needed for both linear and nonlinear cases. This problem can be solved by Ensemble Kalman Filter [41] as in Algorithm 2. In EnKF, the first three equations are predicting equations. In these three equations, an ensemble of *a priori* estimates are simulated based on Eq. (2.7), where E is the size of this ensemble set, a sample mean and a sample covariance are subsequently calculated by this ensemble set. The last three steps are updating equations, they inherit the similar formulations from KF, the *a posteriori* estimate is also a weighted sum of predicted *a priori* estimates and measurement innovation, in which the weight K_k is depends on Q and R.

2.2 Background of Topological Data Analysis

In the fast development of machine learning recent years, TDA has become increasingly popular as a powerful tool in many areas. Researchers have used TDA to solve many real-world problems. A great deal of TDA applications have been developed including signal identification [50], materials science [28, 54], shape recognition [7, 43], histology image

Algorithm 2 Ensemble Kalman filter

$$
\begin{aligned}
\hat{x}_k^{\prime(e)} &= \Psi(\hat{x}_{k-1}^{(e)}) + u_{k-1}^{(e)},\ e = 1, ..., E \\
\hat{m}_k &= \frac{1}{E}\sum_{e=1}^{E} \hat{x}_k^{\prime(e)} \\
\hat{C}_k &= \frac{1}{E-1}\sum_{e=1}^{E} (\hat{x}_k^{\prime(e)} - \hat{m}_k)(\hat{x}_k^{\prime(e)} - \hat{m}_k)^T \\
K_k &= \hat{C}_k H^T (H\hat{C}_k H^T + R)^{-1} \\
\hat{x}_k &= (I - K_k H)\hat{m}_k + K_k z_k \\
\hat{x}_k^{(e)} &= \hat{x}_k + v_k^{(e)},\ e = 1, ..., E
\end{aligned}
$$

analysis [2, 63, 78], ecology of human mobility [14, 15], and cosmology [82, 87]. A review of TDA and its application is provided in [89]. In our work, we use topological nerve in the linking process of tracking algorithm, and then classify actin networks based on their persistence diagrams, therefore we aim to introduce necessary background knowledge of TDA in this section.

2.2.1 Simplicial complexes

In TDA, we need to build a structure which reveals geometric features hidden in the data. We construct this structure by simplicial complexes. Simplcial complexes provide a bridge between the data space and a topological space in which computation of distances between sets of data points can be realized. We start with the definition of simplices.

Definition 2.1. *Let $v_0, v_1, ..., v_k$ be $k+1$ linear independent vertices in $\mathbb{R}^d$. A k-simplex is the set of convex combinations of these $k+1$ vertices,*

$$
s(v_0, v_1, ..., v_k) = \left\{ \sum_{i=0}^{k} \alpha_i v_i \mid \sum_{i=0}^{k} \alpha_i = 1, \alpha_i \geq 0 \right\}.
$$

The faces of a simplex is the all convex combinations in a subset of its vertices.

In particular, higher dimensional simplices are constructed from lower dimensional simplices. From its definite, vertices are 0-simplices. A 1-simplex is called an edge and

is created by its two vertices as faces. Note that a higher dimensional edge is constructed from lower dimensional points. A 2-simplex or a triangle has three edges as faces. Further more, a 3-simplex or a tetrahedron has four triangles as faces.

Definition 2.2. *A simplicial complex is a finite collection of simplices of different dimensions such that faces of simplices are also simplices, and intersections of the simplices are either empty or a face of both.*

An example of a simplicial complex is exhibited in Fig. 2.1. Note that this simplicial complex in 3D is a collection of vertices, edges, a triangle and a tetrahedron, and their intersections are either a face or empty.

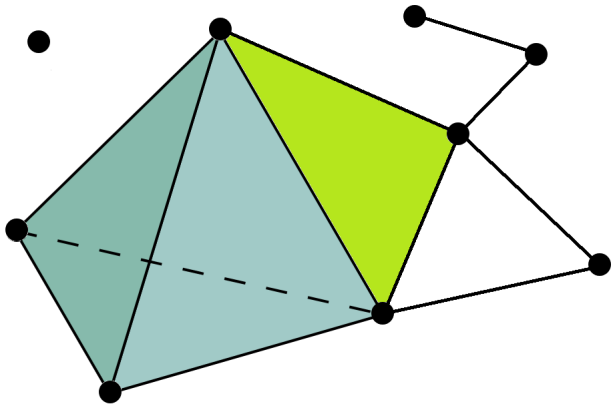

Figure 2.1: A simplicial complex example in $\mathbb{R}^3$. It is a collection of vertices, edges, a triangle and a tetrahedron.

With those definitions and examples above, then we are ready to see how to construct two certain types of simplicial complexes.

Definition 2.3. *Let F be a finite collection of open sets. A nerve $\mathcal{N}(F)$ of F is simplicial complex such that a k-simplex $s(v_0, v_1, ..., v_k)$ is in $\mathcal{N}(F)$ if and only if* $\bigcap_{F_i \in F, i \in \{0,...,k\}} F_i \neq \emptyset$.

Building a nerve is the most basic way to construct a simplicial complex. Different choices of the open sets in F lead to various kinds of simplicial complex. In our intracellular

tracking work, we partition organelle trajectories into small open segments, trajectories are then reconstructed by building a nerve based on intersections of these open segments.

Given the data $\{x_i\}$, if we let F be the finite collection of balls $\{B(x_i, \epsilon)\}$, where $B(x_i, \epsilon)$ is a ball centered at x_i with radius ϵ, than the nerve is called the Čech complex of the data. In the real world, the drawback of the Čech complexes is that they are expensive to compute. Computing the entire complex requires exponential time in the size of X, which is extremely inefficient. Therefore another simplicial complexes called Vietoris-Rips complex [20] has been wildly used instead. Its definition is given as follows,

Definition 2.4. *Let $X = \{x\}$ be a finite set of points in $\mathbb{R}^d$. The Vietoris-Rips complex of X and ϵ is*

$$VR_\epsilon = \{s \subseteq VR_\epsilon | diam\ s \leq 2\epsilon\}\,.$$

In general, in order to build Čech complexes or Vietoris-Rips complexes on a dataset, we just introduce ϵ-balls with radius ϵ and centered at each data point. A simplicial complexes is constructed based on intersections of these ϵ-balls. Every ϵ value is corresponding to a simplicial complex of data points, and various ϵ may cause different complexes. Several Vietoris-Rips complexes examples with different ϵ are exhibited in Fig. 2.2. Five data points in $\mathbb{R}^2$ are given in panel (a). When $\epsilon = 0.5$ in panel (b), none of the ϵ-balls intersect, the Vietoris-Rips complex just contains five vertices. In panel (c), four ϵ-balls in the upper half area have intersected when $\epsilon = 0.71$, thus these four vertices are connected to its contiguous neighbor vertices, and the Vietoris-Rips complex contains four edges and five vertices. When $\epsilon = 0.85$, the Vietoris-Rips complex is built up by five edges and five vertices in panel (d). In the last panel (e), when ϵ is increased to 1, the four ϵ-balls in the upper half area are pairwise intersected, therefore, the Vietoris-Rips complex is constructed by four triangles, seven edges and five vertices.

2.2.2 Persistence diagrams

In this section, we start from briefly introducing homology group with its complementary knowledge.

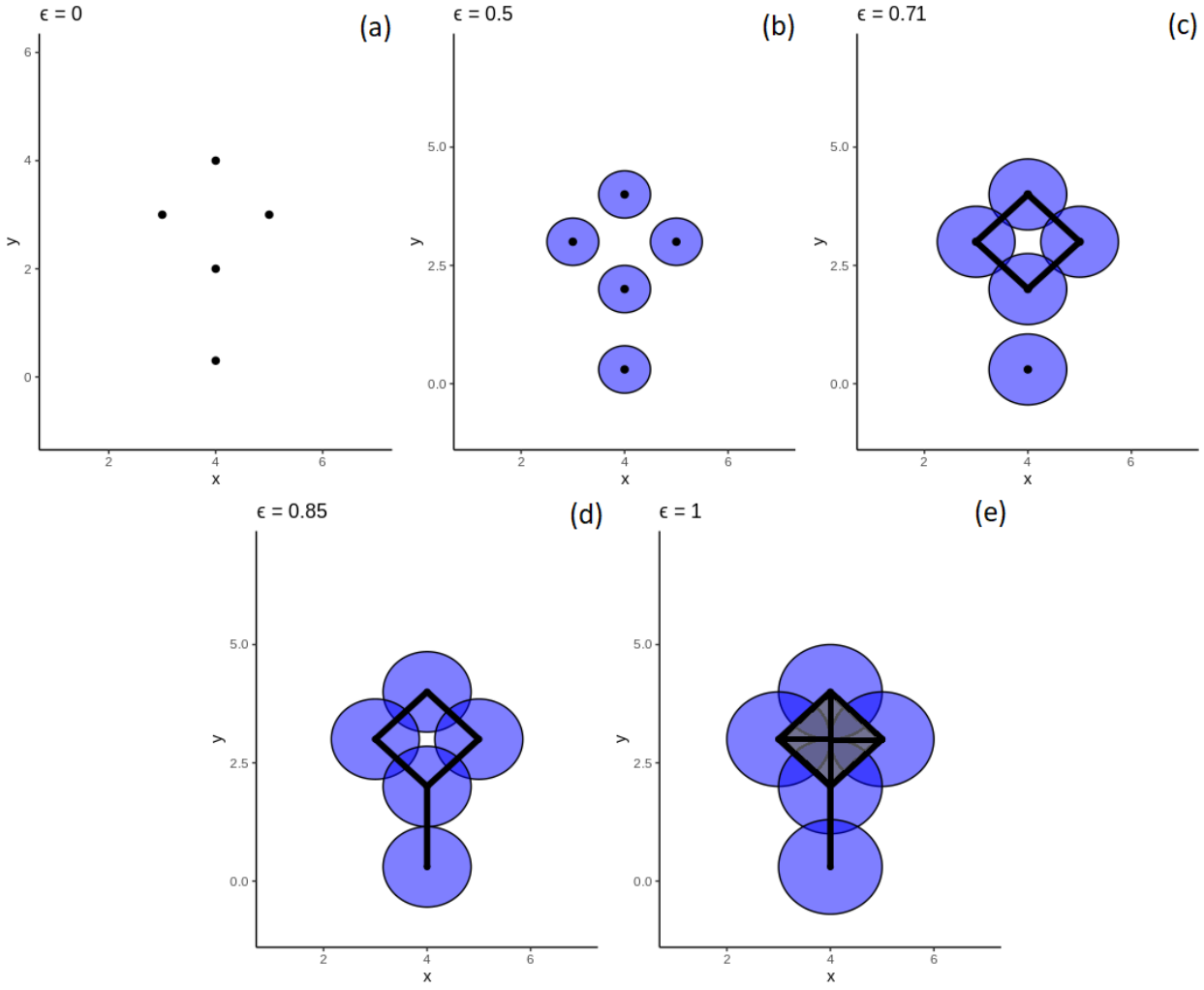

Figure 2.2: This figure shows four Vietoris-Rips complexes (VR_ϵ) of five data points in $\mathbb{R}^2$ with different ϵ values. Note that every ϵ is corresponding to a complex, and diversifying ϵ may cause the change of complexes.

Definition 2.5. *For a simplicial complex $\mathcal{K}$, A p-chain is a formal sum of p-simplices in $\mathcal{K}$, i.e.,*

$$\left\{ \sum_{s_i \in \mathcal{K}} \alpha_i s_i | \alpha_i \in \mathbb{Z}_2 = \{0, 1\},\ s_i \text{ is a } p\text{-simplex} \right\}.$$

A set of p-chains form a p-chain group C_p.

Definition 2.6. *The boundary map is a group homomorphism $\partial : C_p \mapsto C_{p-1}$ such that*

$$\partial s(v_0, v_1, ..., v_k) = \sum_{i=0}^{k} (-1)^i s(v_0, ..., v_{i-1}, v_{i+1}, ..., v_k).$$

Notice that, the boundary of a p-simplex is a sum of its $(p-1)$-dimensional faces, taking boundary of a p-chain will yield a $(p-1)$-chain as the result. For example, the boundary of an edge is its two vertices, the boundary of a triangle is three edges, the boundary of a tetrahedron is three triangles, and so on.

With the boundary map, we define homology groups,

Definition 2.7. *The p-dimensional homology group is the quotient group given by*

$$H_p = \ker(\partial_p)/im(\partial_{p+1}).$$

From the definition, the coset equivalence can be written as $\forall c \in \ker(\partial_p),\ c \sim \{c + c' | c' \in \text{im}(\partial_{p+1})\}$. That means given $c_1 \in \ker(\partial_p)$, $c_1 \sim c_2$ if and only if $c_2 = c_1 + \text{im}(\partial_{p+1})$, which also means c_1, c_2 are differed by a image of ∂_{p+1}. Therefore one specific p-dimensional homology group is corresponding to one type of homological (topological) features in a simplicial complex. In fact, the 0-dim homology group is corresponding to connected components, the 1-dim homology group is corresponding to holes, the 2-dim homology group is corresponding to voids, and generally, the p-dimensional homology group is corresponding to p-spheres in a complex [53]. Further, in our convention, every simplicial complex has a group of homological features, we use Fig. 2.2 as examples. In panel (b), the Vietoris-Rips complex only has five components (0-dim homological feature); the complex in panel (c) has two connected components and one hole (1-dim homological feature); while there are one

connected component and one hole in panel (d); and the complex has only one connected component in the last panel (e).

Once we capture a group of homological features in a simplicial complex (e.g. VR_ϵ) of a dataset with a specific ϵ, moreover, we could get multiple groups of homological features in difference complexes built on the same data with various ϵ, we need a tool to summarize all these information. Persistence diagram is well qualified for this job by visualizing the "birth" and "death" of a homological feature as x-, y-coordinates of a point in the diagram. To construct a persistence diagram for a set of data points, we adopt the procedure of forming Vietoris-Rips complexes by introducing a sequence of ϵ-balls with increasing radius ϵ and centered at each data point. Each value of ϵ yields an unordered group of homological features. Considering values of ϵ as a timeline, we only record when a homological feature appears and disappears. These indexes are called the birth time and death time of a particular homological feature. Moreover, the lifespan (death minus birth) of a homological feature is referred to as the feature's persistence. A set of homological features gives rise to a set of persistence measurements. At the end of this procedure, when radius ϵ is sufficient larger so that the homology group remains unchanged even by further increasing the radius, information of persistent homology (the set of persistent homology measurements) is summarized in a persistence diagram.

We continuous using the five data points in Fig. 2.2 to show the process of building a persistence diagram, and provide a depiction in Fig. 2.3. Given the data points in panel (a), let's investigate ϵ from 0 to 1.25 and only consider 0-dim and 1-dim homological features, which are also known as connected components and holes. When ϵ is relatively small, five individual components are present. When ϵ increases to 0.71, four components merge into one connected component, this simultaneously gives rise to a hole. When ϵ is enlarged to 0.85, all components merge into one connected component, though the hole survives. When ϵ is continuously increased to 1, the only 1-dim hole vanishes. No extra information is gained while ϵ is grown to 1.25. Recording the birth time and death time of a homological feature as x-, y-coordinates, respectively, the appearances and disappearances of 0-dim and 1-dim features are summarized as black dots (correspond to connect components) and red triangles (correspond to holes) in a persistence diagram in panel (f). Notice for 0-dim homological

features, three connected components are born at 0 and die at 0.71, one connected component is born at 0 and dies at 1, one connected component is born at 0 and survive to the end. For 1-dim homological features, one hole is born at 0.71 and dies at 1. Thus, the persistence diagram provides information of connectedness and holes induced from the Vietoris-Rips complexes on the points cloud.

Overall, persistent homology indirectly summarizes the hidden shape of the data and transcribe these shapes to the persistence diagrams. With the persistence diagrams of each point cloud, a classifier can be generated either from the distance [51] between persistence diagrams, or by alternative vectorizations of the diagrams, such as persistence landscape technique [9], persistence image technique [1] and distance statistics technique [52].

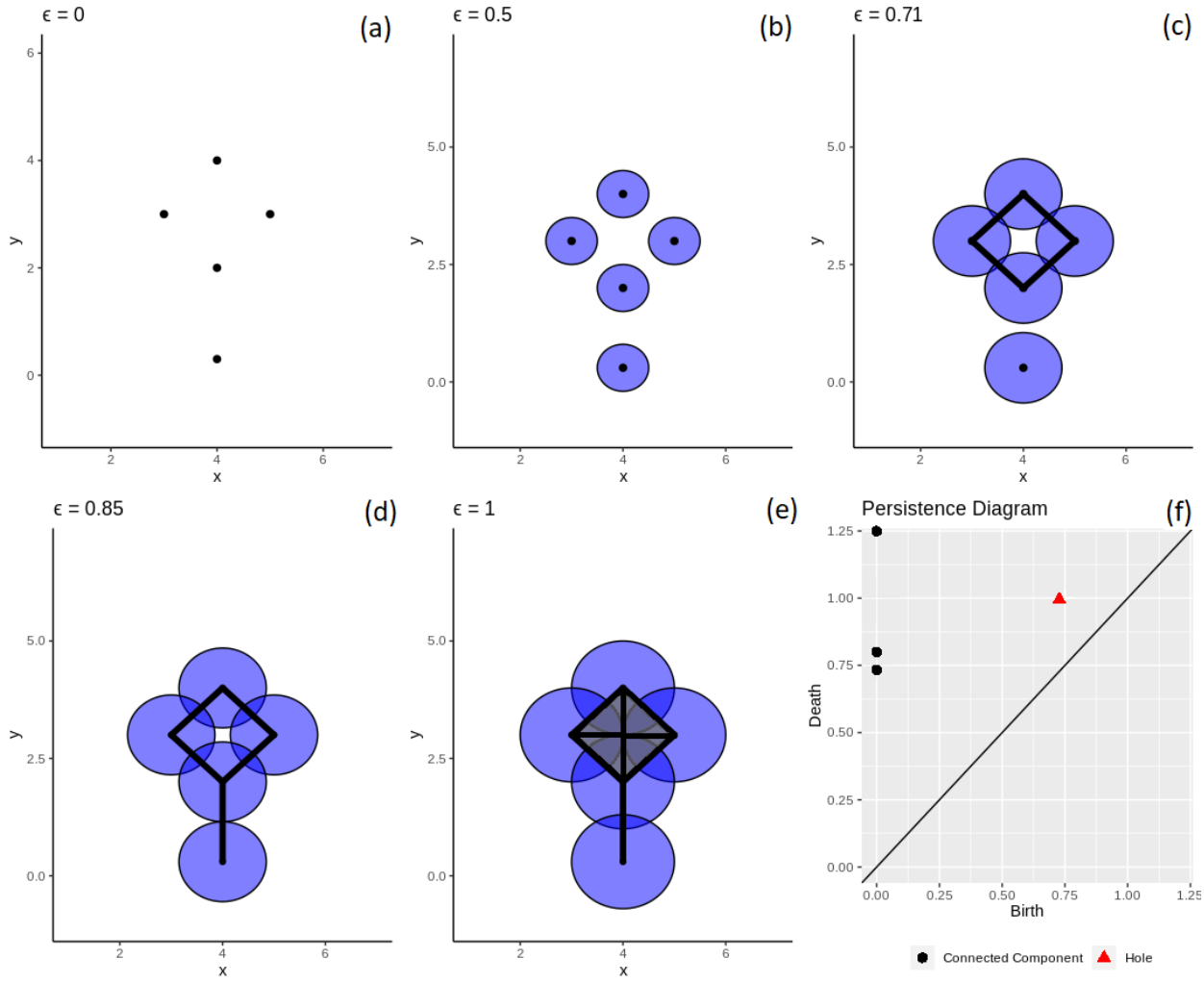

Figure 2.3: Approach depiction of making persistence diagram. Beginning from $\epsilon = 0$, there are five components in Panel (a). When $\epsilon = 0.5$, five components stay alive in Panel (b). In Panel (c), when $\epsilon = 0.71$, four components connect into one connected component, forming a hole, thus three components die and one hole is born. In Panel (d), when $\epsilon = 0.85$, the two components further merge into one, therefore, one more component dies. Finally, when $\epsilon = 1$, the only one hole dies but the only one connected component survives in Panel (e). No additional homological change while ϵ is grown to 1.25. Panel (f) summarizes the persistent homology as a persistence diagram.

Chapter 3

Topological tracking algorithm of sub-cellular motions

3.1 Description of datasets

Datasets that capture the motion of organelles through conventional fluorescence microscopy are typically provided in a video format [27, 42, 76]. Essentially, each video-dataset consists of a stack of pixelated images $\mathcal{D} = \{\mathcal{F}_n, t_n\}_{n=1}^{N}$, where each image $\mathcal{F}_n$ is obtained at time t_n during the course of an experiment.

Ignoring imaging artifacts caused by finite frame rate, dead time, or rolling shutter [42] that are insignificant on the time- and space-scales involved in plant microscopy [45, 47], we consider images obtained at time levels $\{t_n\}_n$ that start at the experiment's onset and end with the experiment's conclusion, denoted $t_1 = 0$ and $t_N = T$, respectively. Further, we consider intermediate time levels that remain equidistant $t_n = (n-1)\Delta t$, where $\Delta t = T/(N-1)$ is the exposure period used for the acquisition of the images.

In turn, each image $\mathcal{F}_n$ is an array of intensity values $\{I_n^p\}_{p=1}^{P}$, where I_n^p denotes the intensity [29, 81], recorded at time t_n of a pixel located at a fixed position $x^p \in \mathbb{R}^2$. We assume that the positions of the pixels $\{x^p\}_{p=1}^{P}$ are given and that they are reported in physical units in the same coordinate system as the sample under imaging. Ignoring positioning parallel to the optical axis (i.e., ignoring z-depth), which is not captured in conventional fluorescence

microscopy [30, 42, 44], we consider $\mathbb{R}^2$ as a plane perpendicular to the optical axis (for example, without any loss of generality, $\mathbb{R}^2$ can model the focal- or xy-plane).

Depending upon the imaging equipment employed in the experiment (e.g., cameras or other light detectors), intensities may be reported in various forms such as photon or electron counts, voltages, currents, ADU (Analog Digital Unit), etc [29, 31, 34, 46, 81]. In this study we assume $\{I_n^p\}_{n,p}$ are given in normalized gray scale values, i.e., I_n^p are measured in arbitrary units (a.u.), with the convention that lower intensities correspond to darker pixels and *vice versa* higher intensities correspond to brighter pixels.

To initiate our method, we model each intensity I_n^p as consisting of a background signal B_n^p, the signal produced by the organelles in the sample J_n^p, and noise n_n^p. That is, we model I_n^p as

$$I_n^p = B_n^p + J_n^p + n_n^p.$$

To find the locations of organelles, we adopt part of the data preprocessing steps and the Bayesian localization step in [75], briefly summarized in the following. In plant microscopy, typically the background signal changes smoothly across the frames. Therefore, we model it as a smooth quadratic surface over the entire field of view and remove it by least square fitting. Next, we model the organelle signal as a sum of Gaussian intensity peaks

$$J_n^p = \sum_{s=1}^{\tilde{S}_n} \tilde{h}_n^s \exp\left(-\frac{\|x^p - \tilde{x}_n^s\|^2}{2(\tilde{w}_n^s)^2}\right),$$

where each peak, labeled by s, is produced by a single organelle [72] that is imaged with maximum intensity $\tilde{h}_n^s > 0$, width $\tilde{w}_n^s > 0$, and center $\tilde{x}_n^s \in \mathbb{R}^2$. We obtain the total number of organelle peaks $\tilde{S}_n$, present in each time level t_n, through an iterative method in which we remove the largest intensity peaks from the frame without the background unitl a certain threshold is met; while we obtain the organelle features $\{(\tilde{x}_n^s, \tilde{h}_n^s, \tilde{w}_n^s)\}_{s=1}^{\tilde{S}_n}$ through the maximum *a posteriori* estimates [13, 24] of a Bayesian model that assumes: (i) the noises $\{n_n^p\}_p$ are independent and Gaussian; (ii) the organelles are *a priori* uniformly positioned over the imaged plane; and (iii) the maximum intensities and widths are *a priori* distributed over appropriate finite intervals.

Ignoring imaging artifacts that are caused by intra-frame motion, which are insignificant in plant microscopy [45, 47], we model each localization $\tilde{x}_n^s \in \mathbb{R}^2$, as the *effective position* of a single organelle at time t_n. In other words, following the localization procedure above, we obtain a collection of space-time positions $\tilde{\mathcal{R}} = \{\{(\tilde{x}_n^s, t_n)\}_{s=1}^{\tilde{S}_n}\}_n \subset \mathbb{R}^2 \times [0, T]$ that reveals the positions of every organelle in the sample only at the experimental time levels $\{t_n\}_n$, see Fig. 3.1.

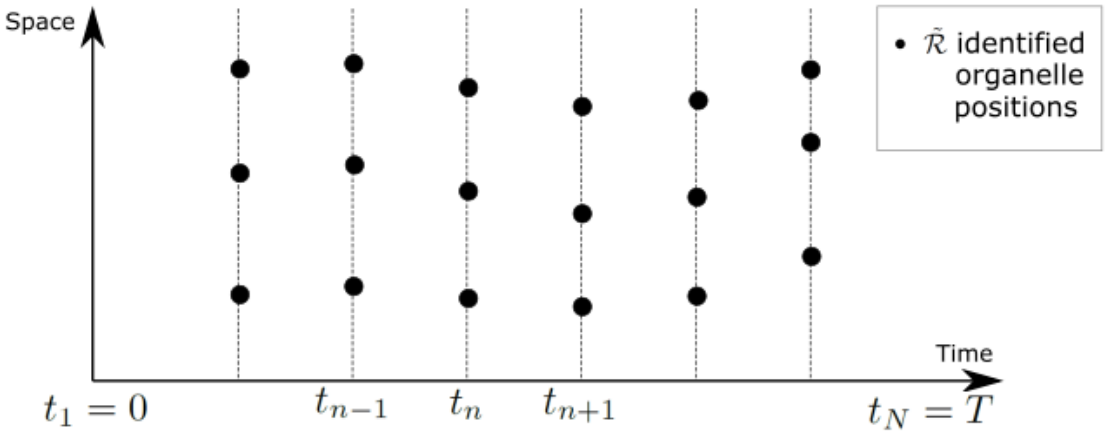

Figure 3.1: The motion of organelles, during an experiment starting at $t_1 = 0$ ending at $t_N = T$, is identified at discrete times t_n (dots). For simplicity, space is represented with one dimension, although real datasets are two dimensional. The black dots represent the locations of organelles at different time levels. $\tilde{\mathcal{R}}$ is the set contains the locations of all black dots.

To proceed with the analysis, we model each organelle's effective position as an idealized point and its motion as a 2D trajectory ignoring motion parallel to the optical axis, which is not captured in a typical dataset. Thus, each organelle, labeled by a, in our formulation corresponds to a *continuous* function $r_a : [0, T] \mapsto \mathbb{R}^2$, where $[0, T]$ represents the time course of the experiment and $\mathbb{R}^2$ represents any plane compatible with the pixel positions $\{x^p\}_p \subset \mathbb{R}^2$. Given a dataset $\mathcal{D}$ of raw experimental observations and a collection of organelle space-time localizations $\tilde{\mathcal{R}}$ identified as described above, our main objective from now on will be the computational reconstruction of $\{r_a\}_a$.

3.2 Ensemble Kalman velocimetry

The motion of the entire set of organelles in the experiment can be encoded within a family of displacement fields $f_{t\to t'}(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}^2$, which we use (see Sect. 3.3, below) to distinguish space-time positions that are visited by each organelle. According to our convention, for any organelle, its position $x, x' \in \mathbb{R}^2$ at a times $t, t' \in [0, T]$ respectively are coupled by the displacement fields as

$$x' = x + f_{t\to t'}(x),$$
$$x = x' + f_{t'\to t}(x').$$

At the spatiotemporal scales probed by conventional fluorescence microscopy, organelles follow irreversible dynamics [60]. Accordingly, the fields $f_{t\to t'}(\cdot)$ and $f_{t'\to t}(\cdot)$ are generally uncorrelated. So, below we incorporate such lack of correlation by adopting a formulation with different forward and backward fields instead of a formulation using only a single field for both temporal directions.

In general, the driving dynamics of organelle motion are unknown, thus the precise form of the fields $\{f_{t\to t'}(\cdot)\}_{t,t'}$ is unknown as well. Next, we describe a method to estimate these fields directly from the raw images in $\mathcal{D}$. For our purpose, it is sufficient to compute displacement fields only at successive time levels. In particular, we are only interested in *1-level forward* $f_{n,+}(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}^2$ and *1-level backward* $f_{n,-}(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}^2$ fields, defined by

$$f_{n,+}(\cdot) = f_{t_n\to t_{n+1}}(\cdot), \qquad n = 1, \ldots, N-1$$
$$f_{n,-}(\cdot) = f_{t_n\to t_{n-1}}(\cdot), \qquad n = 2, \ldots, N.$$

This convention is illustrated in Fig. 3.2.

We compute the displacement fields following a velocimetric approach. We first compute displacements $\{\{\bar{f}^j_{n,+}\}_{n=1}^{N-1}, \{\bar{f}^j_{n,-}\}_{n=2}^{N}\}_{j=1}^{J} \subset \mathbb{R}^2$ at the discrete time levels t_n, for $n = 1, \ldots, N$, of the images in the dataset $\mathcal{D}$ and arbitrarily selected discrete positions $\{\bar{x}^j\}_{j=1}^{J} \subset \mathbb{R}^2$. In particular, given a selected position $\bar{x}^j$, we compute the displacements $\bar{f}^j_{n,+}, \bar{f}^j_{n,-}$ by image registration method between a sub-region of pixels, centered around $\bar{x}^j$, in image $\mathcal{F}_n$ and the

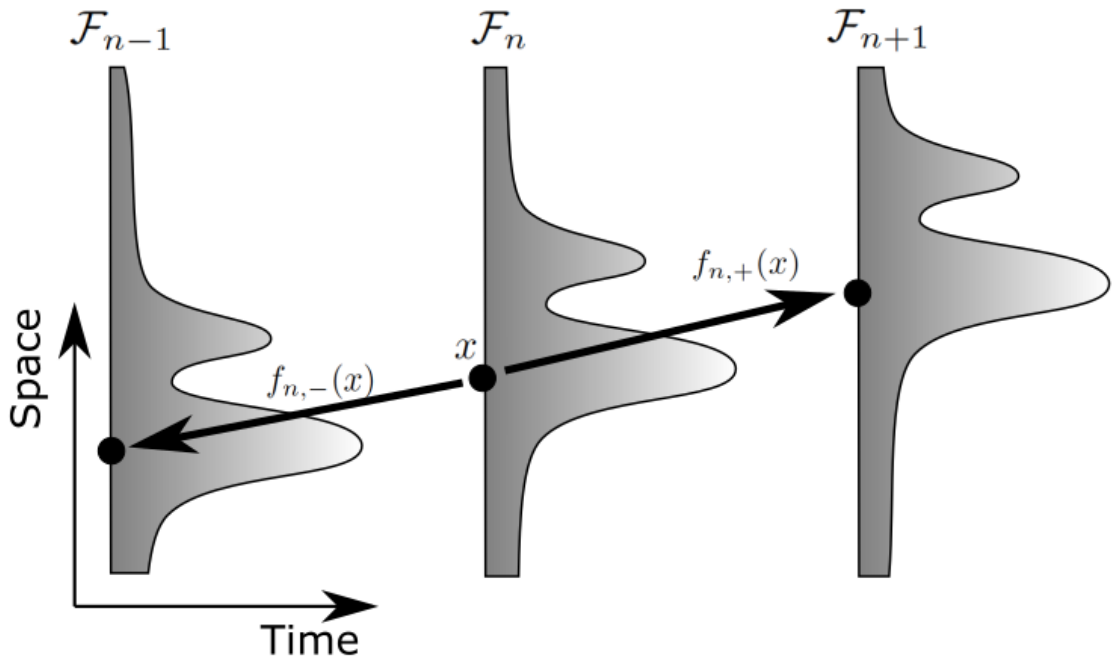

Figure 3.2: Here, x is the position of an organelle producing the images shown (gray), $f_{n,+}(x)$ and $f_{n,-}(x)$ illustrate 1-level forward displacement and backward displacement of x, respectively. For clarity, the image produced by the organelle are shown as multi-peaked and space as 1D.

images $\mathcal{F}_{n+1}$, $\mathcal{F}_{n-1}$, respectively. Briefly, we consider a transformation $T_{\delta,\theta} : \mathbb{R}^2 \mapsto \mathbb{R}^2$ that translates by $\delta \in \mathbb{R}^2$ and rotates by an angle $\theta \in [0, 2\pi)$. Further, we consider $\bar{A}^j$ gathering all pixels $\mathcal{P}$ such that $\|\bar{x}^j - x^{\mathcal{P}}\|_\infty \leq w_{max}$, where $w_{max} > 0$ is a parameter controlling the side length of the region under registration, and set to a small multiple of the typical organelle size. The image registration reduces to solving the following minimization problems

$$\bar{f}^j_{n,+} = \arg\min_{\delta} \left[\min_{\theta} \sum_{\mathcal{P} \in \bar{A}^j} \left| I^{\mathcal{P}}_n - I^{T_{\delta,\theta}(x^{\mathcal{P}})}_{n+1} \right|^2 \right], \qquad j = 1, \ldots, J, \quad n = 1, \ldots, N-1, \tag{3.1}$$

$$\bar{f}^j_{n,-} = \arg\min_{\delta} \left[\min_{\theta} \sum_{\mathcal{P} \in \bar{A}^j} \left| I^{\mathcal{P}}_n - I^{T_{\delta,\theta}(x^{\mathcal{P}})}_{n-1} \right|^2 \right], \qquad j = 1, \ldots, J, \quad n = 2, \ldots, N. \tag{3.2}$$

Additionally, to exclude arbitrarily large displacements, we restrict each minimization over only displacements $\|\delta\| \leq d_{max}$, where $d_{max} > 0$ is an upper bound on the longest distance an organelle can travel during one exposure Δt.

To extend our discrete displacements over the entire $\mathbb{R}^2$ support, obtain globally defined displacement fields, and account for the errors introduced in prediction, we adopt a representation of the *forward field*

$$f_{1,+}(\cdot) = u_{1,+}(\cdot), \tag{3.3}$$

$$f_{n,+}(\cdot) = \Psi_+(f_{n-1,+}(\cdot)) + u_{n,+}(\cdot), \quad n = 2, \ldots, N-1 \tag{3.4}$$

and a similar representation for the *backward field*

$$f_{n,-}(\cdot) = \Psi_-(f_{n+1,-}(\cdot)) + u_{n,-}(\cdot), \quad n = 2, \ldots, N-1 \tag{3.5}$$

$$f_{N,-}(\cdot) = u_{N,-}(\cdot), \tag{3.6}$$

where $\Psi_+(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}^2$ and $\Psi_-(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}^2$ describe how the displacement fields change from one frame to its immediate ancestor and predecessor. $\Psi_+(\cdot), \Psi_-(\cdot)$ could be motion equations of a dynamic system if it was known or just ansatzes based on previous experience. There is a special case when $\Psi_+(\cdot), \Psi_-(\cdot)$ are identity functions, this happens when one trusts the displacement fields reserve the same trend as the previous level. The driving noise $\{u_{n,+}(\cdot)\}_{n=1}^{N-1}$ and $\{u_{n,-}(\cdot)\}_{n=2}^{N}$ are independent Gaussian processes with mean zero and covariances that correlates the x or y components of the displacement fields according to a kernel $K(\cdot,\cdot) : \mathbb{R}^2 \times \mathbb{R}^2 \mapsto (0, \infty)$. To facilitate the computations, we leave the x and y components independent from each other. To ensure smooth fields that do not change rapidly across organelles, we use the squared exponential kernel

$$K(x, x') = \sigma_u^2 \exp\left(-\frac{1}{2}\left(\frac{\|x - x'\|}{\ell}\right)^2\right), \tag{3.7}$$

where $\sigma_u^2 > 0$ is a constant and we set $\ell > 0$ approximately equal to the diameter of a single organelle. Here σ_u^2 presents the credibility of prediction system, i.e., if $\Psi_+(\cdot), \Psi_-(\cdot)$ are quite believable then σ_u^2 should be chosen to be small, vice versa, if $\Psi_+(\cdot), \Psi_-(\cdot)$ are uninformative then σ_u^2 should be relatively large.

Due to imperfections in image registration, the displacements $\{\{\bar{f}^j_{n,+}\}_{n=1}^{N-1}, \{\bar{f}^j_{n,-}\}_{n=2}^{N}\}_j$ computed through Eqs. (3.1),(3.2) may deviate from the true displacements $\{\{f_{n,+}(\bar{x}^j)\}_{n=1}^{N-1}, \{f^j_{n,-}(\bar{x}^j)\}_{n=2}^{N}\}_j \subset \mathbb{R}^2$ at the corresponding positions $\bar{x}^j$. To account for such errors, we combine these fields with the displacements $\{\{\bar{f}^j_{n,+}\}_{n=1}^{N-1}, \{\bar{f}^j_{n,-}\}_{n=2}^{N}\}_j$ to form a noisy phenomenological observation model

$$\bar{f}^j_{n,+} = f_{n,+}(\bar{x}^j) + v^j_{n,+}, \qquad n = 1, \ldots, N-1 \tag{3.8}$$

$$\bar{f}^j_{n,-} = f_{n,-}(\bar{x}^j) + v^j_{n,-}, \qquad n = 2, \ldots, N \tag{3.9}$$

where $\{v_{n,+}\}_{n=1}^{N-1}$ and $\{v_{n,-}\}_{n=2}^{N}$ are independent bivariate Gaussian random variables with zero mean and variances $\sigma_v^2 > 0$. Here σ_v^2 measures the reliability of observed displacements $\{\{\bar{f}^j_{n,+}\}_{n=1}^{N-1}, \{\bar{f}^j_{n,-}\}_{n=2}^{N}\}_j$ acquired from image registration method, smaller σ_v^2 indicates closer agreement with the true displacement.

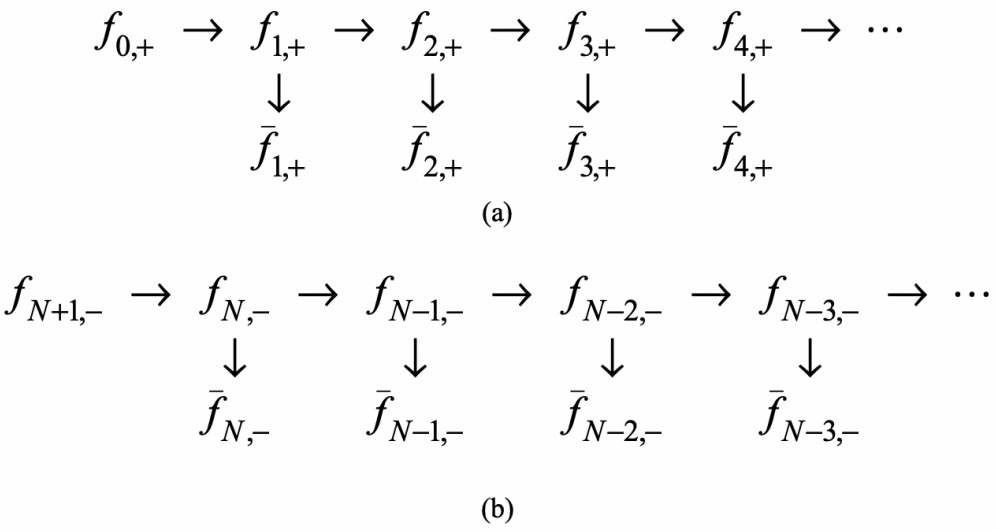

Figure 3.3: The relations of forward fields and backward fields are indicated here. (a) shows the approach depiction of forward displacement fields, (b) shows the approach depiction of backward displacement fields. For clarity, time marches forward in (a) and backward in (b).

An implementation of Eqs. (3.3)-(3.6), which apply in continuous space, is computationally intractable yielding to a pertinent discretization. Precisely, we apply a grid of

fixed positions $\{\bar{\bar{x}}^\lambda\}_{\lambda=1}^{\Lambda} \subset \mathbb{R}^2$ that may not, in general, coincide with $\{\bar{x}^j\}_j$. Next, let $\phi_{n,+}(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}$, with $n = 1, \ldots, N-1$, denote the x component of the displacement field $f_{n,+}(\cdot)$.

Denoting $\left[\phi_{n,+}(\bar{\bar{x}}^1) \ \cdots \ \phi_{n,+}(\bar{\bar{x}}^\Lambda)\right]^T$ by $\Phi_{n,+}(\bar{\bar{X}}^\Lambda)$ and $\left[\phi_{n,+}(\bar{x}^1) \ \cdots \ \phi_{n,+}(\bar{x}^J)\right]^T$ by $\Phi_{n,+}(\bar{X}^J)$, then according to the Gaussian process $u_{1,+}(\cdot)$, Eq. (3.3) becomes

$$\left[\Phi_{1,+}(\bar{\bar{X}}^\Lambda) \quad \Phi_{1,+}(\bar{X}^J)\right]^T \sim N_{\Lambda+J}\left(0_{(\Lambda+J)\times 1}, \Sigma\right), \tag{3.10}$$

where $\Sigma = \left[\begin{array}{ccc|ccc} K(\bar{\bar{x}}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^1,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{x}^J) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^\Lambda,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{x}^J) \\ \hline K(\bar{x}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^1,\bar{\bar{x}}^\Lambda) & K(\bar{x}^1,\bar{x}^1) & \cdots & K(\bar{x}^1,\bar{x}^J) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ K(\bar{x}^J,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^J,\bar{\bar{x}}^\Lambda) & K(\bar{x}^J,\bar{x}^1) & \cdots & K(\bar{x}^J,\bar{x}^J) \end{array}\right]$.

Similarly, according to the Gaussian processes $\{u_{n,+}(\cdot)\}_{n=2}^{N-1}$, Eq. (3.4) becomes

$$\left[\Phi_{n,+}(\bar{\bar{X}}^\Lambda) \quad \Phi_{n,+}(\bar{X}^J)\right]^T \sim N_{\Lambda+J}\left(\Psi_+\left(\left[\Phi_{n-1,+}(\bar{\bar{X}}^\Lambda) \quad \Phi_{n-1,+}(\bar{X}^J)\right]^T\right), \Sigma\right), \tag{3.11}$$

where $n = 2, \ldots, N-1$.

Let $\bar{\phi}_{n,+}^j \in \mathbb{R}$ denote the x component of $\bar{f}_{n,+}^j$ and further denote the vector $\left[\bar{\phi}_{n,+}^1 \ \cdots \ \bar{\phi}_{n,+}^J\right]^T$ by $\bar{\Phi}_{n,+}^J$. Then Eq. (3.8) becomes

$$\bar{\Phi}_{n,+}^J \sim N_J\left(\left[0_{J\times\Lambda} \quad I_{J\times J}\right]\left[\Phi_{n,+}(\bar{\bar{X}}^\Lambda) \quad \Phi_{n,+}(\bar{X}^J)\right]^T, \sigma_v^2 I_{J\times J}\right), \qquad n = 1, \ldots, N-1. \tag{3.12}$$

Analogous formulas apply for the y component $\psi_{n,+}(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}$ of the forward field, as well as for the x component $\phi_{n,-}(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}$ and y component $\psi_{n,-}(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}$ of the backward field. We provide the complete set of equations in the Appendix.

We consider the Ensemble Kalman Filtering (EnKF) [41] to compute the posterior point estimates. Let notation $[\cdot^j]$ denote the vector, which contains all elements corresponding

possible value j. In Eqs. (3.10),(3.11), $\{[\phi^j_{n,+}]\}_{n=1}^{N-1}$ are the states in $\mathbb{R}^{\Lambda+J}$, in which predictions are made based on $\Psi_+(\cdot)$(see Eqs. (3.4),(3.11)), and $\{[\bar{\phi}^j_{n,+}]\}_{n=1}^{N-1}$ are the states in $\mathbb{R}^J$, in which the observations are obtained by applying image registration to the image data. With Eqs. (3.3),(3.4),(3.8) and their vectorized discrete form (3.10),(3.11),(3.12), EnKF will produce a sequence of improved estimations $\{[\hat{\phi}^j_{n,+}]\}_{n=1}^{N-1}$ for the x component of forward fields.

To adopt the EnKF algorithm for our problem, we mainly perform predicting and updating steps iteratively. Denote $Q = \Sigma$ and $R = \sigma_v^2 I_{J\times J}$, and let E be the size of an ensemble we choose. In every iteration, instead of using a single estimation, EnKF generates an ensemble of samples based on multiple predictions $\hat{\phi}^{(e)}_n$ in the following equation,

$$\hat{\phi}^{(e)}_n = \Psi_+(\phi^{(e)}_{n-1}) + [u^j_{n,+}]^{(e)},\ [u^j_{n,+}]^{(e)} \sim N(0, Q),\ e = 1, ..., E,$$

where one sample is corresponding to one simulation satisfying Eq. (3.11). Sample mean and sample variance needed in the following updating steps are computed subsequently by this ensemble set as follows,

$$\hat{m}_n = \frac{1}{E}\sum_{e=1}^{E} \hat{\phi}^{(e)}_n$$
$$\hat{C}_n = \frac{1}{E-1}\sum_{e=1}^{E} (\hat{\phi}^{(e)}_n - \hat{m}_n)(\hat{\phi}^{(e)}_n - \hat{m}_n)^T.$$

Then the Kalman gain denoted by G_n is calculated by $G_n = \hat{C}_n(\hat{C}_n + R)^{-1}$. It controls the weight of the predictions $\hat{\phi}^{(e)}_n$ and observation $[\bar{\phi}^j_{n,+}]$ to be involved in our approximation $[\hat{\phi}^j_{n,+}]$, where the improved estimation $[\hat{\phi}^j_{n,+}]$ is obtained by updating the sample mean of predictions with the observation in the following way,

$$[\hat{\phi}^j_{n,+}] = (I - G_n)\hat{m}_n + G_n[\bar{\phi}^j_{n,+}]. \tag{3.13}$$

Actually, the improved estimation $[\hat{\phi}^j_{n,+}]$ is a weighted sum of predictions $\hat{\phi}^{(e)}_n$ and observation $[\bar{\phi}^j_{n,+}]$ depends on σ_u^2 the credibility of prediction system and σ_v^2 the reliability of observation. Suppose the observation is more reliable, meaning $\sigma_v^2 < \sigma_u^2$ and $\sigma_v^2 \to 0$, then $\lim_{\sigma_v^2\to 0} G_n = I$,

and $\lim_{\sigma_v^2 \to 0}(I - G_n) = 0$, hence $[\hat{\phi}_{n,+}^j]$ in Eq. (3.13) has less information from the mean of predictions $\hat{m}_n$, but contains more information from the observation $[\bar{\phi}_{n,+}^j]$, therefore the observation holds a heavier weight in the improved estimation. Conversely, suppose the prediction process is more trustworthy, or equivalently, $\sigma_u^2 < \sigma_v^2$ and $\sigma_u^2 \to 0$, the improved estimation weights the predictions more heavily [5]. At the end of every iteration, samples in the resemble set are further updated as

$$\phi_n^{(e)} = [\hat{\phi}_{n,+}^j] + [v_{n,+}^j]^{(e)}, \ [v_{n,+}^j]^{(e)} \sim N(0, R), \ e = 1, ..., E.$$

Thus, after all iterations, a sequence of improved estimations $\{[\hat{\phi}_{n,+}^j]\}_{n=1}^{N-1}$ is obtained by applying EnKF. All steps of EnKF algorithm for organelle velocimetry is summarized in Algorithm 3.

The y component of forward fields and backward fields filtering process works in a similar way. With the discreterized equations given in Appendix and the approach depiction in Fig. 3.3, one could perform the EnKF to obtain the improved estimations $\{[\hat{\psi}_{n,+}^j]\}_{n=1}^{N}$, $\{[\hat{\phi}_{n,-}^j]\}_{n=1}^{N}$ and $\{[\hat{\psi}_{n,-}^j]\}_{n=1}^{N}$.

3.3 Topological reconstruction

Given a collection of organelle space-time localizations $\tilde{\mathcal{R}}$ and appropriate displacement fields $\{f_{n,+}(\cdot)\}_{n=1}^{N-1}$ and $\{f_{n,-}(\cdot)\}_{n=2}^{N}$, our goal is to computationally reconstruct $\{r_a\}_a$. Of course, because the reconstruction of $\{r_a\}_a$ in continuous time is impossible without a motion model capable of time interpolation, which is unavailable for plant organelles, we focus on reconstructing trajectories $\{\tilde{r}_a\}_a$ that are discretized at time levels contained in $\tilde{\mathcal{R}}$, i.e., $\tilde{r}_a = \{r_a(t_n)\}_n$. As we show below, for such discrete reconstruction the computed displacement fields $\{f_{n,+}(\cdot)\}_{n=1}^{N-1}$ and $\{f_{n,-}(\cdot)\}_{n=2}^{N}$ are sufficient.

We adopt a similar linking process as in [75]. Our algorithm (described in depth below) proceeds in three stages. See Fig. 3.4 for visual representation. In the *first stage*, we embed

Algorithm 3 Ensemble Kalman Filter for Organelle Velocimetry

$$\begin{aligned}\phi_0^{(e)} &= 0,\ e = 1, ..., E\\ Q &= \Sigma\\ R &= \sigma_v^2 I\end{aligned}$$

When $n = 1$,

$$\begin{aligned}\hat{\phi}_1^{(e)} &= \phi_0^{(e)} + [u_{1,+}^j]^{(e)},\ [u_{1,+}^j]^{(e)} \sim N(0, Q),\ e = 1, ..., E\\ \hat{m}_1 &= \frac{1}{E}\sum_{e=1}^{E} \hat{\phi}_1^{(e)}\\ \hat{C}_1 &= \frac{1}{E-1}\sum_{e=1}^{E}(\hat{\phi}_1^{(e)} - \hat{m}_1)(\hat{\phi}_1^{(e)} - \hat{m}_1)^T\\ G_1 &= \hat{C}_1(\hat{C}_1 + R)^{-1}\\ [\hat{\phi}_{1,+}^j] &= (I - G_1)\hat{m}_1 + G_1[\bar{\phi}_{1,+}^j]\\ \phi_1^{(e)} &= [\hat{\phi}_{1,+}^j] + [v_{1,+}^j]^{(e)},\ [v_{1,+}^j]^{(e)} \sim N(0, R),\ e = 1, ..., E\end{aligned}$$

For $n = 2$ to $N - 1$

$$\begin{aligned}\hat{\phi}_n^{(e)} &= \Psi_+(\phi_{n-1}^{(e)}) + [u_{n,+}^j]^{(e)},\ [u_{n,+}^j]^{(e)} \sim N(0, Q),\ e = 1, ..., E\\ \hat{m}_n &= \frac{1}{E}\sum_{e=1}^{E} \hat{\phi}_n^{(e)}\\ \hat{C}_n &= \frac{1}{E-1}\sum_{e=1}^{E}(\hat{\phi}_n^{(e)} - \hat{m}_n)(\hat{\phi}_n^{(e)} - \hat{m}_n)^T\\ G_n &= \hat{C}_n(\hat{C}_n + R)^{-1}\\ [\hat{\phi}_{n,+}^j] &= (I - G_n)\hat{m}_n + G_n[\bar{\phi}_{n,+}^j]\\ \phi_n^{(e)} &= [\hat{\phi}_{n,+}^j] + [v_{n,+}^j]^{(e)},\ [v_{n,+}^j]^{(e)} \sim N(0, R),\ e = 1, ..., E\end{aligned}$$

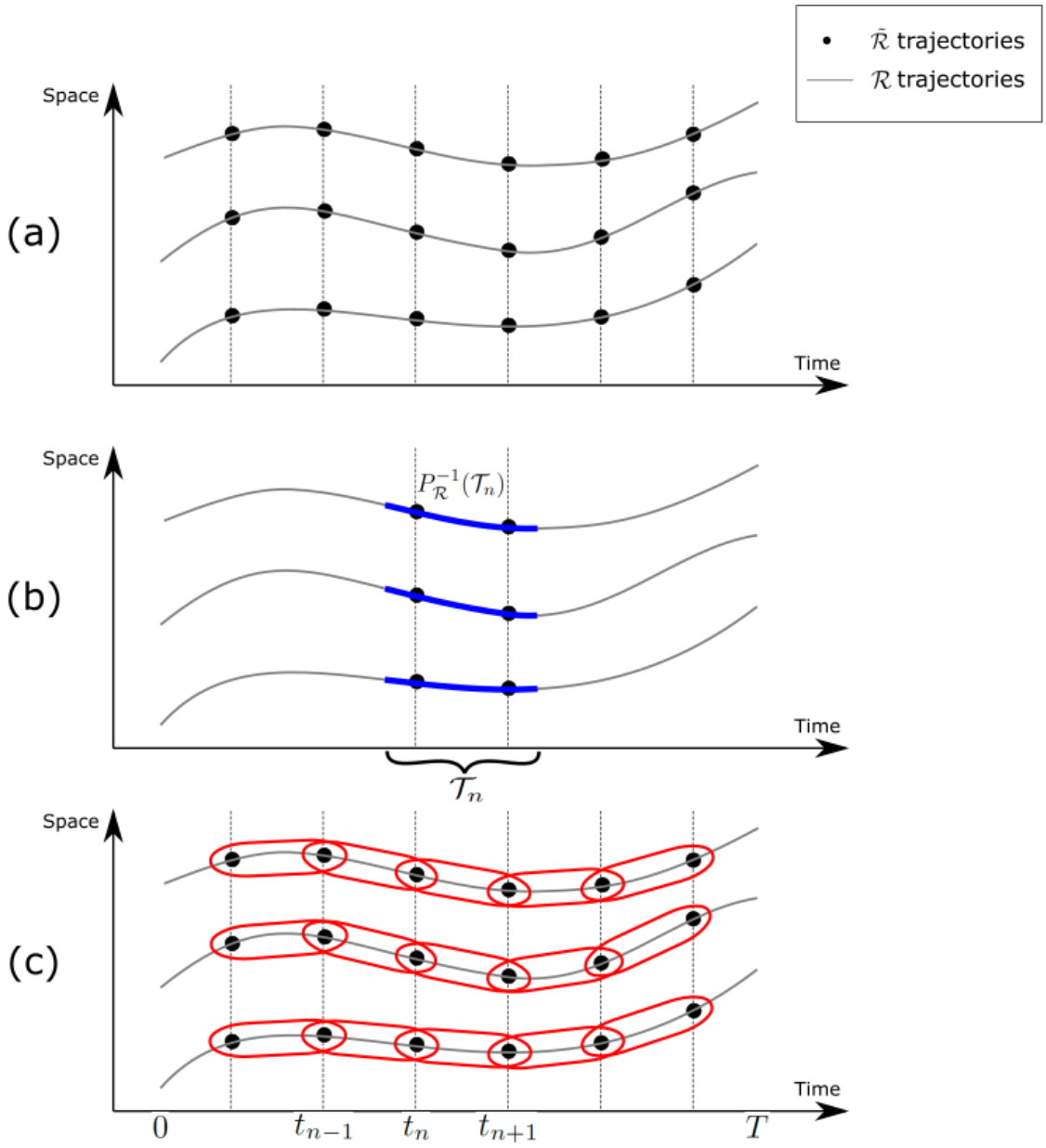

Figure 3.4: (a) shows $\tilde{\mathcal{R}}$ as black dots and $\mathcal{R}$ as gray lines; (b) shows $\mathcal{R}$, $\mathcal{T}_n$ in Eq. (3.14) and $P_{\mathcal{R}}^{-1}(\mathcal{T}_n)$ as blue segments; (c) shows $\mathcal{R}$, $\mathcal{T}_n$, $P_{\mathcal{R}}^{-1}(\mathcal{T}_n)$, $\tilde{R}$ and reconstructed discrete trajectories. For visualization purpose, space is shown in 1D.

$\tilde{\mathcal{R}}$ into

$$\mathcal{R} = \bigcup_a \{ (r_a(t), t) \}_{t\in[0,T]} \subset \mathbb{R}^2 \times [0, T].$$

Then for any two points $(x,t) \in \mathbb{R}^2 \times [0,T]$ and $(x',t') \in \mathbb{R}^2 \times [0,T]$, we consider

$$d\left((x,t),(x',t')\right) = \|x - x'\| + \alpha|t - t'|,$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^2$ and $\alpha > 0$ is constant. Since $d(\cdot,\cdot)$ is a distance in $\mathbb{R}^2 \times [0,T]$, our main object of interest $\mathcal{R} \subset \mathbb{R}^2 \times [0,T]$ inherits the topological properties of a metric space [21, 57, 83]. Essentially, $\mathcal{R}$ consists of the points in space-time $\mathbb{R}^2 \times [0,T]$ that are visited by the organelles during the experiment. Although $\mathcal{R}$ globally captures the motion we are interested in revealing, it leaves individual trajectories indistinguishable. Accordingly, in the *second stage*, we partition $\mathcal{R}$ into components $\{\mathcal{R}_a\}_a$ such that each $\mathcal{R}_a$ corresponds to a single trajectory r_a, i.e., we partition $\mathcal{R} = \cup_a \mathcal{R}_a$ such that $\mathcal{R}_a = \left\{ \left(r_a(t),t\right) \right\}_{t\in[0,T]} \subset \mathbb{R}^2 \times [0,T]$. The partitioning of $\mathcal{R}$ can be computationally achieved through construction of the appropriate topological nerve [20] via the *Mapper algorithm* [12, 77]. Briefly, for any $\tau > 0$, such that $\tau < \Delta t$, we consider the overlapping intervals $\{\mathcal{T}_n\}_{n=1}^{N-1}$ defined by

$$\begin{aligned} \mathcal{T}_1 &= [t_1, t_2 + \tau), \\ \mathcal{T}_n &= (t_n - \tau, t_{n+1} + \tau), \qquad n = 2, \ldots, N-2, \\ \mathcal{T}_{N-1} &= (t_{N-1} - \tau, t_N] \end{aligned} \tag{3.14}$$

which are associated with the time levels $\{t_n\}_n$ of the provided dataset. For any $(x,t) \in \mathcal{R}$ we consider the temporal projection $P_{\mathcal{R}} : \mathcal{R} \mapsto [0,T]$ defined by

$$P_{\mathcal{R}}\left((x,t)\right) = t, \qquad (x,t) \in \mathcal{R}.$$

Due to continuity, $\{P_{\mathcal{R}}^{-1}(\mathcal{T}_n)\}_{n=1}^{N-1} \subset \mathcal{R}$ forms an open covering of $\mathcal{R}$. By its definition, each pre-image $P_{\mathcal{R}}^{-1}(\mathcal{T}_n) \subset \mathcal{R}$ contains segments of at least one organelle trajectory, however, due to its inherited topology, each trajectory segment corresponds to only a single connected component within $P_{\mathcal{R}}^{-1}(\mathcal{T}_n) \subset \mathcal{R}$. Consequently, partitioning $\mathcal{R}$ into connected components is achieved by, first partitioning each $P_{\mathcal{R}}^{-1}(\mathcal{T}_n)$ into its connected components $\{\mathcal{S}_{m,n}\}_{m=1}^{M_n}$ and, computing subsequently the nerve of the entire resulting family of components $\{\{\mathcal{S}_{m,n}\}_{m=1}^{M_n}\}_{n=1}^{N-1} \subset \mathcal{R}$, which is also an open covering of $\mathcal{R}$. Lastly, in the

third stage, we readily obtain discrete trajectories $\tilde{r}_a$ by intersecting $\mathcal{R}_a \cap \tilde{\mathcal{R}}$. To partition each $P_{\mathcal{R}}^{-1}(\mathcal{T}_n)$ into its connected components $\{\mathcal{S}_{m,n}\}_m$, we consider

$$\ell\left((x,t),(x',t')\right) = \|x - x' - f_{t'\to t}(x')\| + \|x' - x - f_{t\to t'}(x)\|,$$

where $(x,t) \in \mathbb{R}^2 \times [0,T]$ and $(x',t') \in \mathbb{R}^2 \times [0,T]$. Here, the points $(x,t) \in \mathbb{R}^2 \times [0,T]$ and $(x + f_{t\to t'}(x), t') \in \mathbb{R}^2 \times [0,T]$ *or* the points $(x',t') \in \mathbb{R}^2 \times [0,T]$ and $(x' + f_{t'\to t}(x'), t) \in \mathbb{R}^2 \times [0,T]$ are both produced by the same organelles. Thus, $(x,t) \in P_{\mathcal{R}}^{-1}(\mathcal{T}_n)$ and $(x',t') \in P_{\mathcal{R}}^{-1}(\mathcal{T}_n)$, belong to the same connected component $S_{m,n}$ *if and only if* $\ell((x,t),(x',t')) = 0$. Therefore, provided the 1-level displacement fields $\{f_{n,+}(\cdot)\}_{n=1}^{N-1}$ and $\{f_{n,-}(\cdot)\}_{n=2}^{N}$ have been already computed, we can use ℓ to topologically characterize trajectory segments or equivalently connected components of $P_{\mathcal{R}}^{-1}(\mathcal{T}_n)$. Consequently, a computational characterization of $\tilde{\mathcal{S}}_{m,n} = \mathcal{S}_{m,n} \cap \tilde{\mathcal{R}}$ can be achieved by an agglomerative clustering on $P_{\mathcal{R}}^{-1}(\mathcal{T}_n) \cap \tilde{\mathcal{R}}$ with linkage ℓ. Specifically, the restriction ℓ_n of ℓ in $P_{\mathcal{R}}^{-1}(\mathcal{T}_n) \cap \tilde{\mathcal{R}}$, required for each clustering, reduces to

$$\ell_n\left((x,t),(x',t')\right) = \begin{cases} \|x - x' - f_{n+1,-}(x')\| + \|x' - x - f_{n,+}(x)\|, & t < t' \\ 2\|x - x'\|, & t = t' \\ \|x - x' - f_{n,+}(x')\| + \|x' - x - f_{n+1,-}(x)\|, & t > t'. \end{cases}$$

3.4 Results

3.4.1 Case I: Velocimetry benchmark

The displacement estimation and linking processes are tested on a simulated data set consisting of 20 organelles in 100 frames of video with a time delay $\Delta t = 1$ s. The trajectories are exhibited in Fig. 3.5. The positions of an organelle in each frame are known and are generated by a diffusion process, which also contains a drift term, given by

$$dX_t = v_x dt + DW_t,$$
$$dY_t = v_y dt + DW_t,$$

where W_t is a Wiener process, $v_x = 3$ pixel/s, $v_y = 1$ pixel/s, and $D = 1$ pixel/s. The starting distances between any two adjacent organelles at $t = 0$ s is 10 pixels.

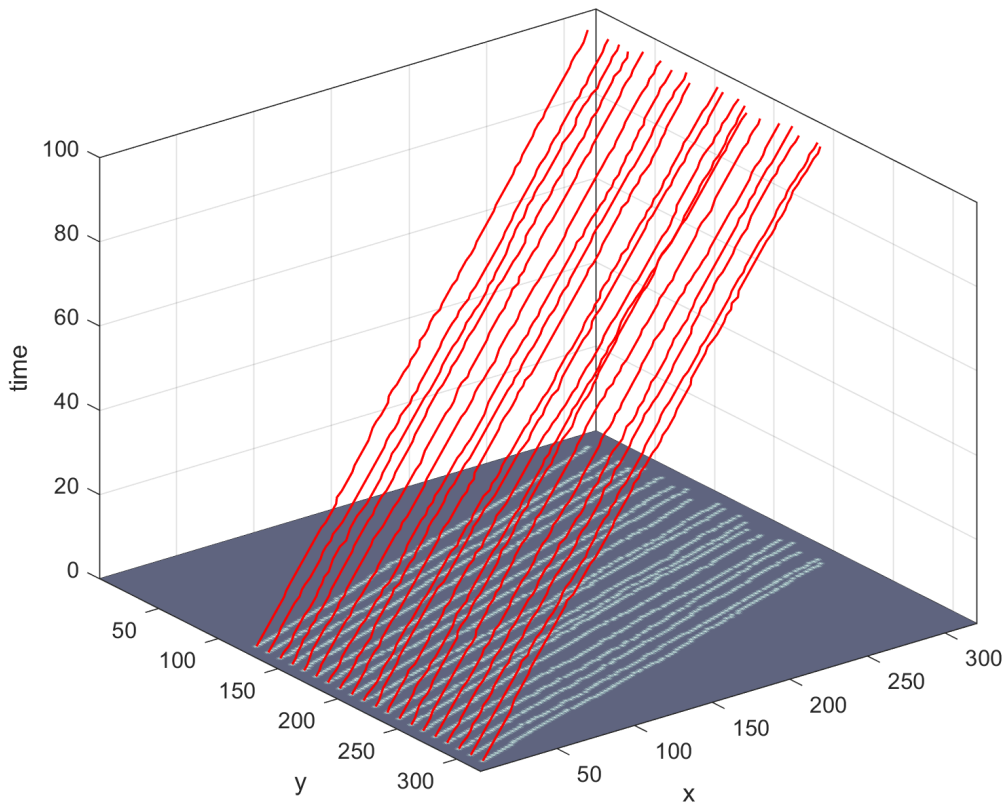

Figure 3.5: Case I: The frame size is 320 by 320 pixels. Trajectories of 20 organelles are in red spanning from time $t = 0$ s to $t = 99$ s. Their motion is described by a diffusion process containing both a diffusion and a drift term. The starting distance of any two adjacent organelles at $t = 0$ s is 10 pixels.

Given the location of all organelles in each frame, we apply our displacement estimation process detailed in Sect. 3.2 to the data set, then calculate the mean error (in pixels) between the estimated forward (backward) displacement and true displacement frame by frame, along the x-axis and y-axis respectively. The results are shown in Fig. 3.6. The four histograms, almost all mean errors per frame are around 0.25 pixel and smaller than one pixel, only very few are greater than one pixel and all are smaller than two pixels, we may see these as outliers.

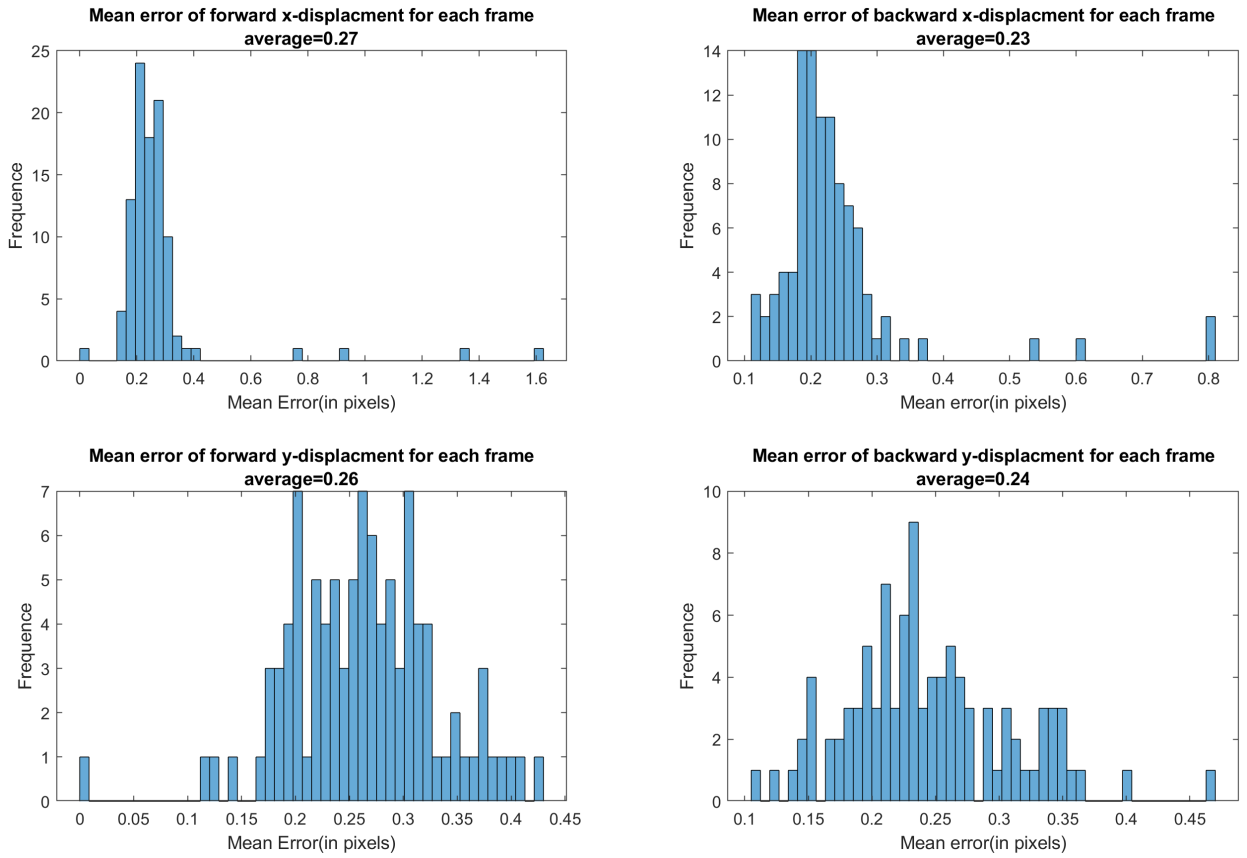

Figure 3.6: Case I: Four histograms of mean error of each frame. Each one compares estimated forward and backward displacement with ground truth along x-axis and y-axis, respectively.

Given the displacements, we apply the linking process of Sect. 3.3, and the results are shown in Fig. 3.7. All organelles are correctly connected by 20 trajectories, each trajectory spans exactly from $t = 0$ s to $t = 99$ s, and the yielding accuracy rate is 100%.

Next we perturb the y-axis direction of the location of simulated organelles, by adding noise following a uniform distribution $U(-\epsilon, \epsilon)$ at every time level, where ϵ is the largest perturbation could be added. We will investigate the cases when ϵ varies from 1 pixel to 4 pixels. Apply the displacement estimation and linking processes with our algorithm, then count number of total reconstructed trajectories, number of reconstructed trajectories longer than 10 s, number of reconstructed trajectories having 100% agreement with the truth, number of reconstructed trajectories having at least 90% agreement with the truth

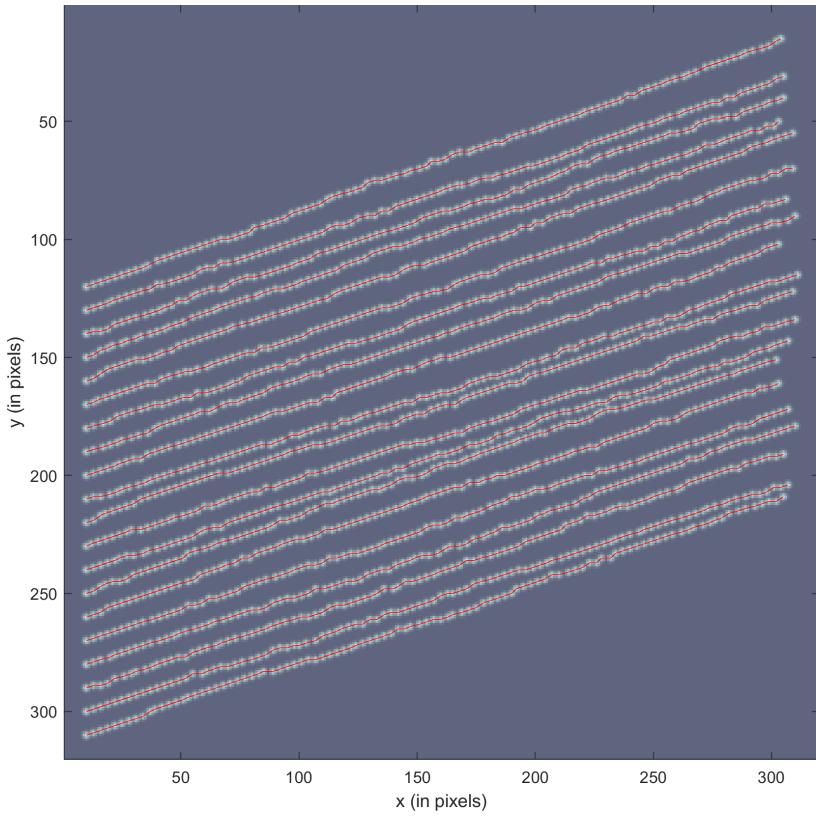

Figure 3.7: Case I: Linking result of all trajectories in red. The accuracy rate is 100%.

and number of reconstructed trajectories having at least 50% agreement with the truth. The results are in Table 3.1.

Table 3.1: Case I: Table of detection result

ϵ **(in pixels)**	**total**	$> 10\ s$	$= 100\%$	$\geq 90\%$	$\geq 50\%$
$\epsilon = 1$	20	20	20	20	20
$\epsilon = 1.5$	20	20	20	20	20
$\epsilon = 2$	20	20	20	20	20
$\epsilon = 2.5$	24	24	14	17	20
$\epsilon = 3$	32	27	9	14	17
$\epsilon = 3.5$	33	30	5	10	15
$\epsilon = 4$	53	40	1	4	14

As shown in Fig. 3.8, when we increase ϵ, the trajectories contain larger fluctuations, and any two adjacent trajectories become closer or even intersect. Thus, larger ϵ causes higher difficulty to detect trajectories. From Table 3.1, when the noise is mild ($\epsilon < 2.5$ pixels), our reconstructed trajectories remain the same; but when ϵ becomes large ($\epsilon \geq 2.5$ pixels), the accuracy rate decreases. In fact, when $\epsilon = 4$ pixels, there are no clear patterns for all independent trajectories to be detected, and most of organelles just look scattered in the frame when overlapping all their positions over the time span of the video.

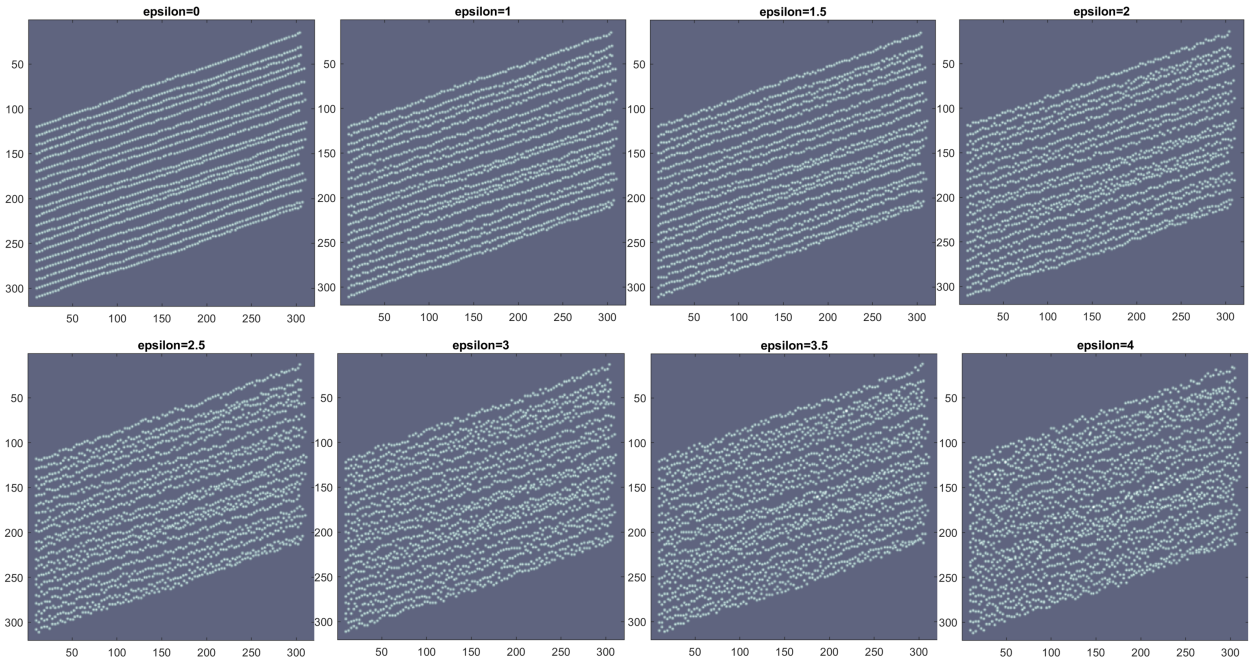

Figure 3.8: Case I: Positions of organelles over time after adding perturbation $U(-\epsilon, \epsilon)$ when $\epsilon = 0,\ 1,\ 1.5,\ 2,\ 2.5,\ 3,\ 3.5,\ 4$ pixels, respectively. If ϵ increases, it is more difficult to detect trajectories, especially, when $\epsilon = 4$ pixels, there are no clear patterns for all trajectories to be reconstructed.

3.4.2 Case II: Complex dynamics

Now consider a complex video with 20 organelles in each frame and 100 frames in total. Each frame has a 380 by 380 pixels grid on it. This video has a frame rate of 30 frames per

second, which gives $\Delta t = 33.33$ ms. There are multiple filaments hiding in the background, and are not visible in the imagery. Three kinds of motions could happen. An organelle could attach to or detach from a filament, travel along a filament, or move randomly. Moreover, an organelle could go through multiple of these three motions in a single Δt.

After importing the video as gray scaled images and filtering out the background from each frame, our method detects peaks iteratively and applies Bayesian identification with the following prior distributions:

- $\tilde{n}_n^p$ follows normal distribution $N(0, 1)$,
- z_n^s follows uniform distribution over the frame,
- h_n^s follows translated beta distribution with support $(50, 150)$, mode 100, and shape parameter $\alpha = 5$,
- w_n^s follows translated beta distribution with support $(10, 20)$, mode 15, and shape parameter $\alpha = 5$,

as mentioned in Sect. 3.1. An example of a single frame is shown in Fig. 3.9. The red dots on the left panel and blue pentagons on the right panel are the original locations before Bayesian identification. It is clear to see that the three pairs at the bottom (which have y-value greater than 300) need to be corrected as part of their corresponding organelles are overlapped. The red pentagons are the fitted location after Bayesian identification.

For the approximation of displacement fields using Ensemble Kalman filtering(see Section 2.2), since $\Delta t = 33$ ms is considered extremely small, this ensures the displacement fields do not change rapidly from one frame to the very next. Moreover, the displacement fields from one level should partially memorize the trend from the previous level. Thus, lacking more information about the dynamic system, we choose $\Psi(x) = \sqrt{x}$ to imitate a nonlinear decay in the displacement field. Setting $\sigma_u = 5$ pixels, $\sigma_v = 2$ pixels since we want to give more weight to observations, then the forward displacement fields of the 17th frame at $t_n = 0.53$ s is displayed in Fig. 3.10(a), an area of pixels $[140, 230]$ on x-axis times pixels $[260, 350]$ on y-axis is enlarged in Fig. 3.10(b). We can easily observe the displacement fields around organelles.

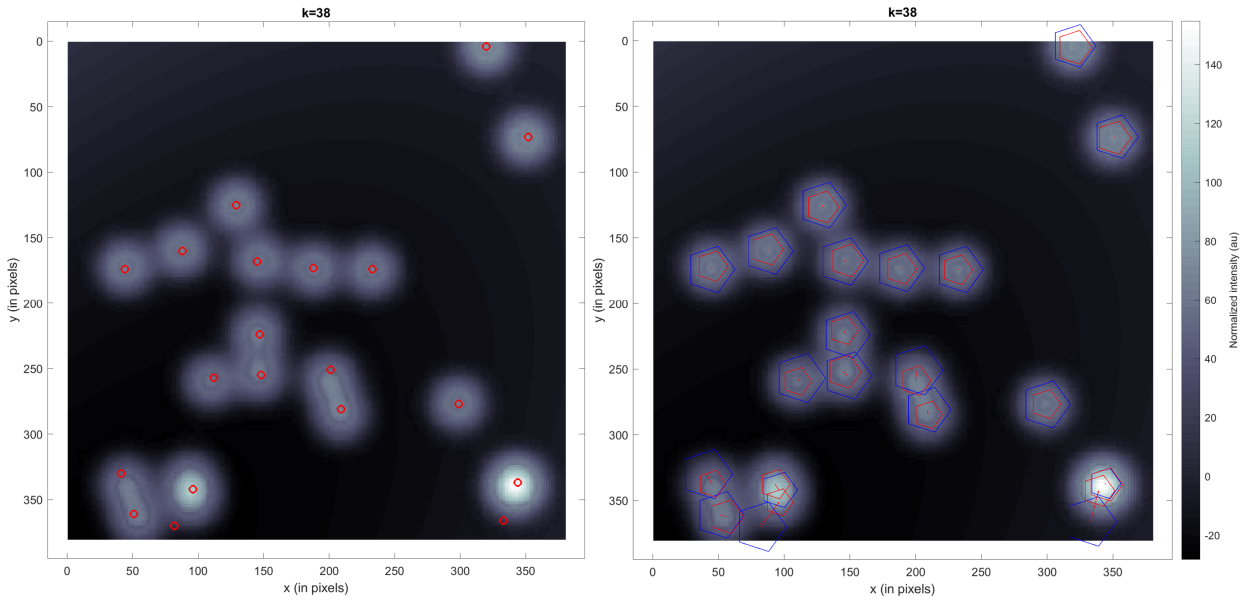

Figure 3.9: Case II: The left shows the rough detection result, the right show the locations after correction. The red dots in the left penal and blue pentagons in the right penal are the original locations before Bayesian identification. The red pentagons in the right panel are the fitted location after Bayesian identification.

The trajectories are reconstructed in Fig. 3.11. The left panel shows all estimated trajectories in red concentrate upon the light area. The right panel shows trajectories with ground truth trajectories in black. Most of them coincide, except the area where our method cannot do the tracking job perfectly when more organelles collide or stick together. Specifically, we pick four sets of trajectory reconstructions, exhibited in Fig. 3.12, each panel shows our reconstructions compared with one true trajectory, their mean error are 1.20, 2.24, 2.09 and 1.42 pixels, respectively.

3.4.3 Case III: Real Data

Finally, we consider a real grayscale video with a total of 299 frames, recording the motion of peroxisomes in a plant cell. In plant cells, peroxisomes play a variety of roles including converting fatty acids to sugar and assisting chloroplasts in photorespiration. The spatial resolution for this video is 0.196 micrometers/pixel and the size of each frame is 79 by 662

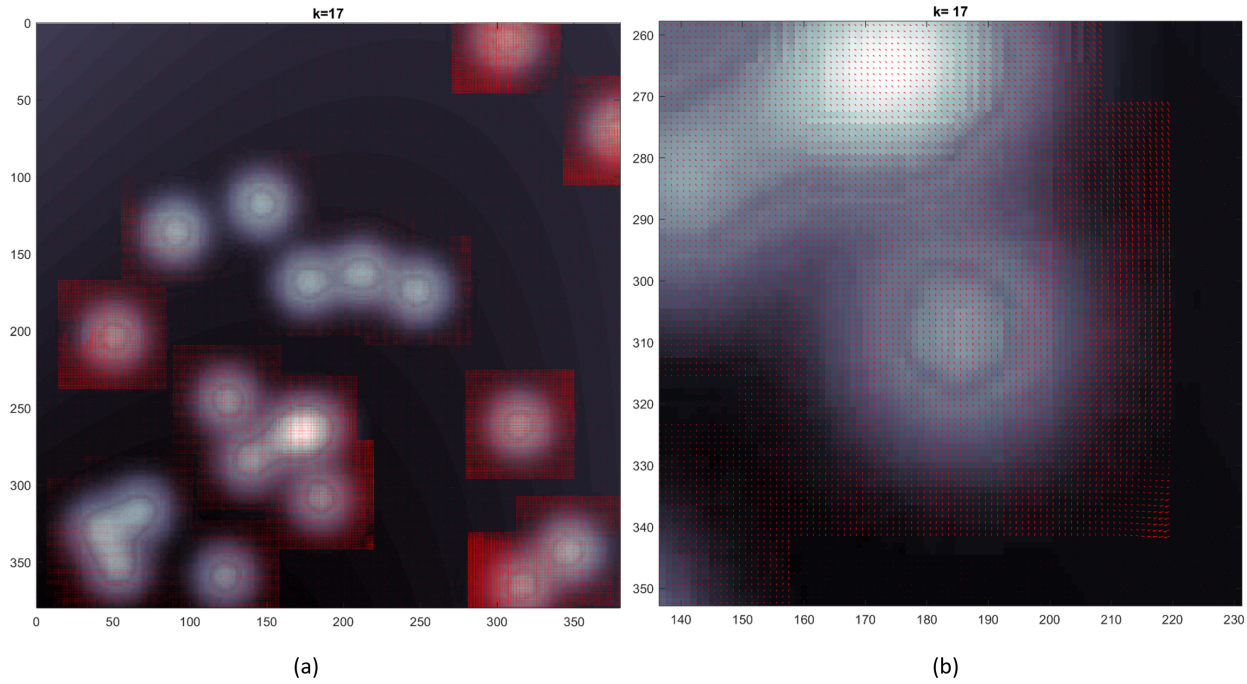

Figure 3.10: Case II: Estimated displacement fields of 17th frame using EnKF. Panel (a) shows the estimated displacement fields for the entire focal plane. Panel (b) shows the enlarged area of $[140, 230] \times [260, 350]$ in Penal (a).

pixels. The time period between successive frames is 82 ms, that is, $\Delta t = 82$ ms. Fig. 3.13(a) shows the first frame of the video, the light spots in the frame are peroxisomes and their sizes range from 0.5 to 1 micrometer.

The outcomes of our method applied to this video are shown in Fig. 3.13(b) and 3.13(c). We plot the estimated trajectories, which only exist in at least 10 consecutive frames, in Fig. 3.13(b). We can see that the red trajectories cover almost every highlighted area. In Fig. 3.13(c), we exhibit all these 116 trajectories in different colors. Mainly two type of trajectories are observed: a long trail when peroxisome is traveling along the filament; a short trail when peroxisome is wiggling in the cell. Our developed method is able to track the peroxisomes in different types of motions for long time intervals.

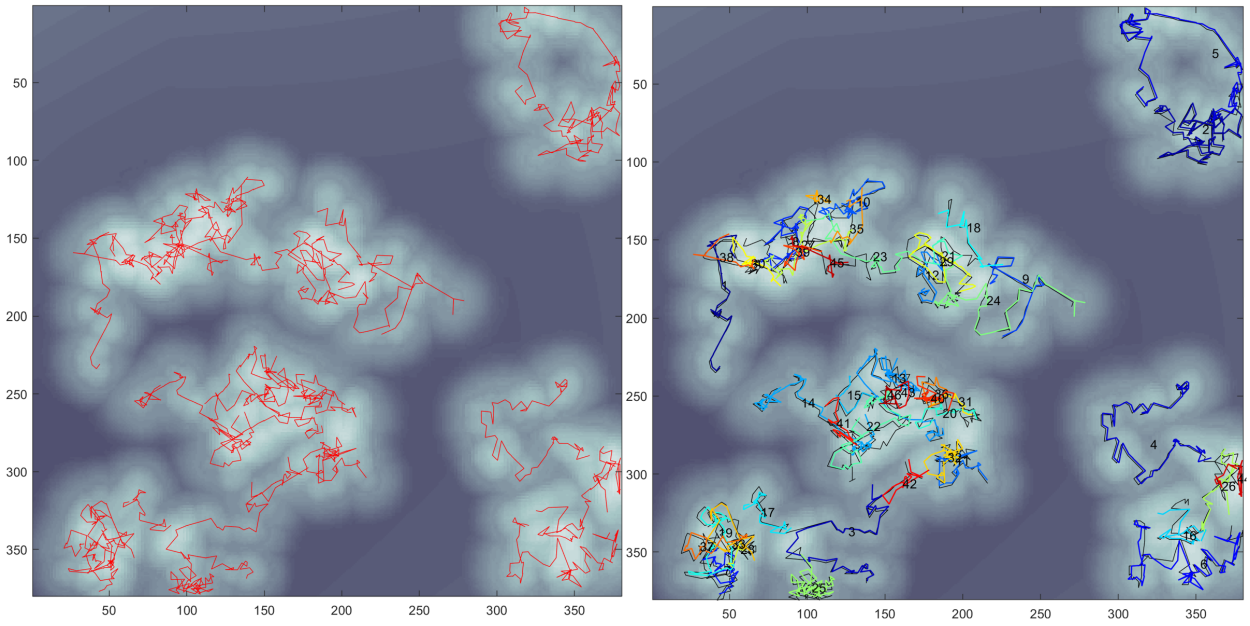

Figure 3.11: Case II: Trajectories reconstruction result

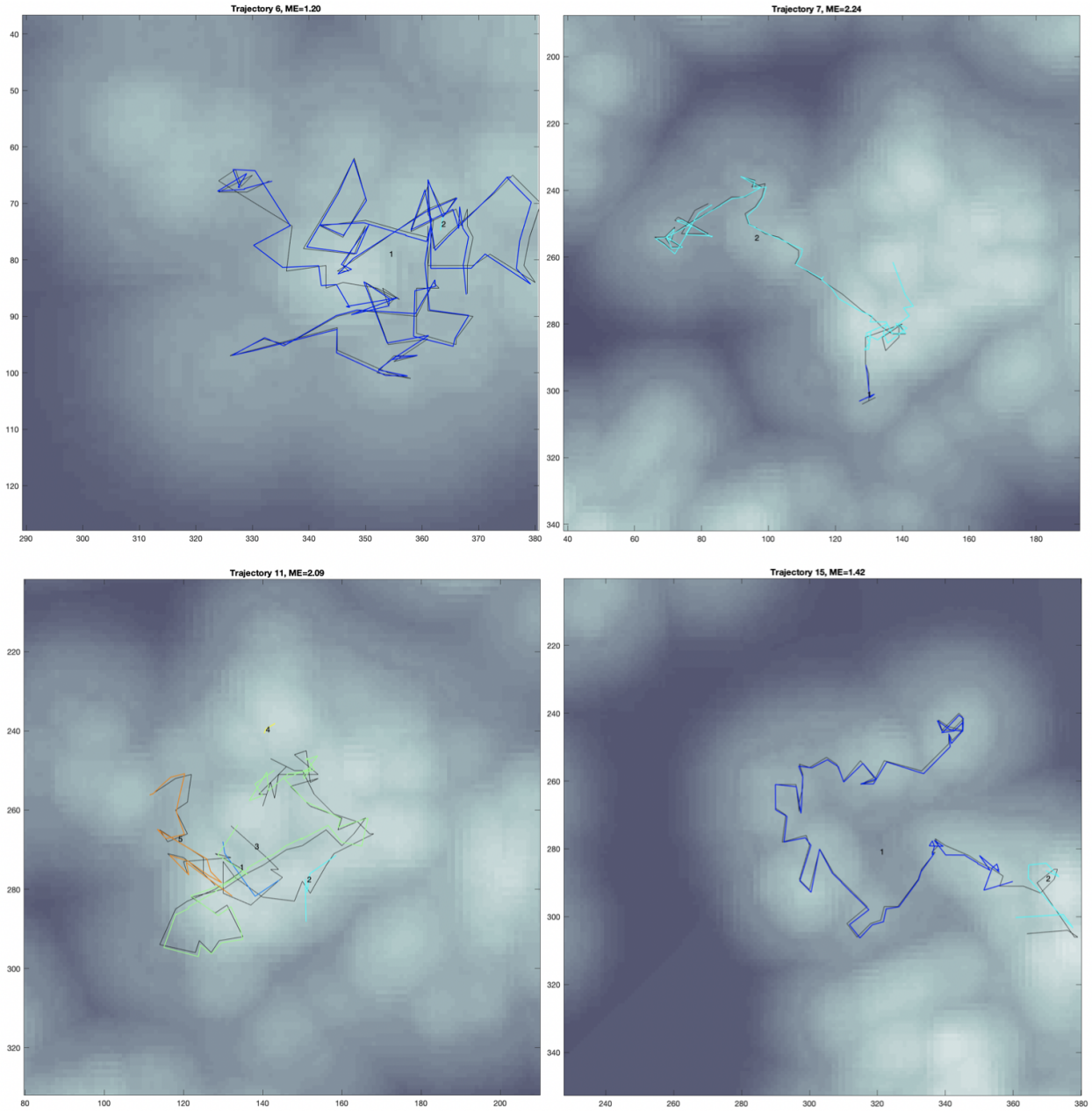

Figure 3.12: Case II: Four specific sets of trajectory reconstructions vs ground truth. Each panel shows reconstructions versus one true trajectory. The upper left is amplified from the area $[290, 380] \times [40, 130]$ in Figure 3.11; the upper right is amplified from the area $[40, 190] \times [190, 340]$ in Figure 3.11; the bottom left is amplified from the area $[80, 210] \times [200, 330]$ in Figure 3.11; the bottom right is amplified from the area $[230, 380] \times [200, 350]$ in Figure 3.11;

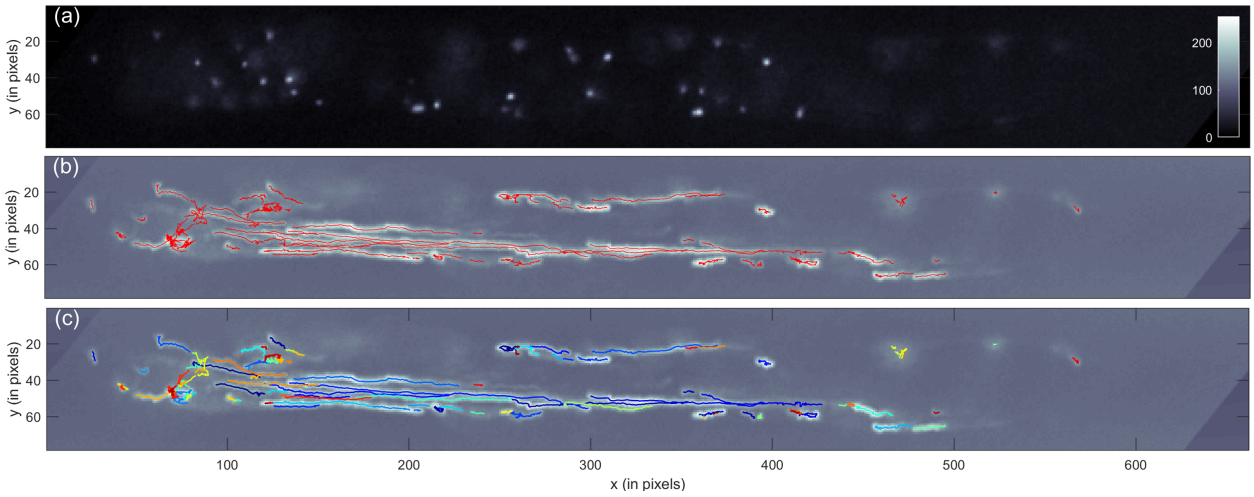

Figure 3.13: Case III: Panel (a) is the first frame of the video. Panel (b) exhibits all estimated trajectories in red. Panel (c) further shows each estimated trajectory in different colors.

Chapter 4

Filament networks learning

4.1 Filament networks and data preprocessing

Actin filaments are thought to be organized by cross-linking on actin-binding proteins [88]. Filaments and inter-filament structure can be simulated by a physical model [22, 23]. However, the change of environment in a eukaryotic cell will cause variation in filament networks. Our network data is simulated by three different cross-linker densities, which are corresponding to three distinct cellular environments. Higher cross-linker density yields more opportunities for filaments to cross-link, i.e. the binding and unbinding processes are more active in a certain area. As shown in Fig. 4.1(a), three kinds of filaments networks are simulated with different numbers of cross-linkers: 825, 1650 and 3300. All cells are bounded by a $20\,\mu m \times 20\,\mu m$ square, and the cross-linking density of each network is 2.06, 4.13 and 8.25 per μm^2, respectively. In each network, there are totally 100 filaments with an average length $10\,\mu m$, they are modeled as polar warm-like chain in red and blue dots represent barbed ends of these filaments. We also record the locations of actin beads that make up the filaments, which are shown as small black circles in Fig. 4.1(b). Every actin bead has an identical radius $0.5\,\mu m$. We study filament networks as a classification problem, and we are interested in developing an automated method to accurately classify cross-linker density of a filament network.

Using the approach of Sect. 2.2.1 and Sect. 2.2.2, we construct simplicial complexes in a typical way of persistent homology by using the 2-dimensional coordinates of actin

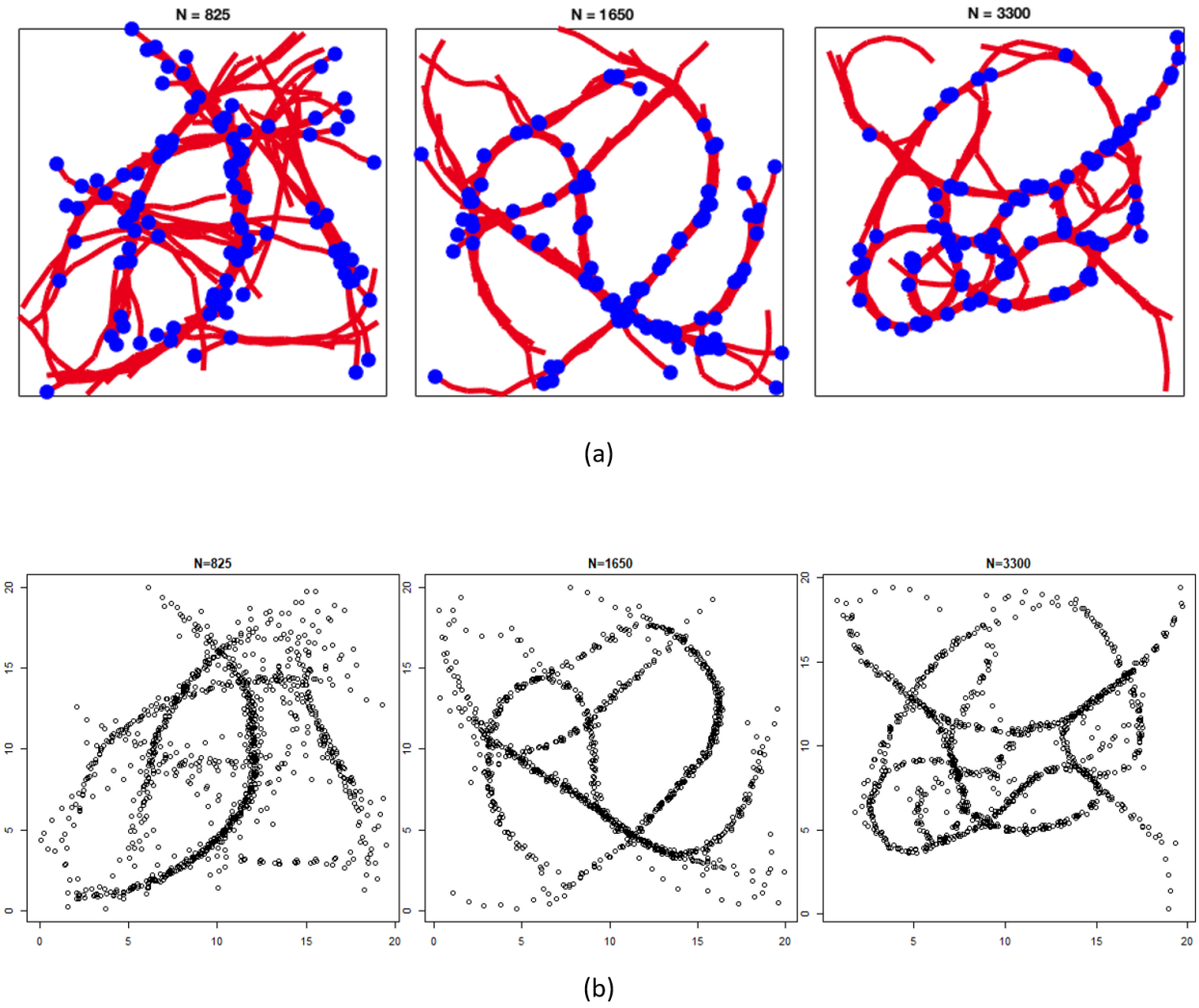

Figure 4.1: Filament networks. Panel (a) shows three filament networks generated by 825, 1650 and 3300 cross-linkers, respectively, in a 20 µm × 20 µm area. Each network contains 100 filaments which are represented as red lines. The blue dots are the barbed ends of these filaments. Panel (b) shows the locations of the actin beads that make up the filaments exhibited in Panel (a).

beads along the filaments as the initial nodes. We adopt the procedure of forming Vietoris-Rips complexes [20] on each dataset (actin network) by introducing a sequence of ϵ-balls with increasing radius ϵ and centered at each data point (actin bead), where each value ϵ corresponds to an unordered group of homological features. We only record when a homological feature appears and vanishes. At the end of this procedure, information of a filament network's persistent homology is summarized in a pertinent persistence diagram.

4.2 Filament network classifier

Once we generate persistence diagrams that correspond to the actin filament networks, we are ready to classify these networks. In this work, we propose a distance-based methodology for a filament network classifier.

Given any two persistence diagrams, we need a way to quantify the difference between them. In TDA, two distances are commonly used, the Bottleneck and the Wasserstein distance [1, 9, 20, 40, 89]. Their definitions are given as follows,

Definition 4.1. *Given two persistence diagrams X and Y. The Wasserstein distence $W_p(X, Y)$ is defined by*

$$W_p(X,Y) = \left(\inf_{\eta: X \mapsto Y} \sum_{x \in X} ||x - \eta(x)||_\infty^p \right)^{\frac{1}{p}}, \tag{4.1}$$

where the infimum is taken over all bijections η. If $p \to \infty$, then Wasserstein distence becomes the Bottleneck distance,

$$W_\infty(X,Y) = \inf_{\eta: X \mapsto Y} \sup_{x \in X} ||x - \eta(x)||_\infty. \tag{4.2}$$

These distances calculate the optimal (minimal) cost in matching points between two persistence diagrams. To ensure the bijection η between persistence diagrams X and Y exists, matching to the diagonal (where birth equals death in persistence diagrams) in the Wasserstein distance and Bottleneck distance is allowed. They assume infinitely many points

of infinite multiplicity on the diagonal. Thus, they only penalize extra points by imposing cost of connecting to the diagonal.

In addition to the Wasserstein and the Bottleneck distance, in this work we adopt a new distance, called d_p^c distance, which is proposed in [51] and has been proved to be stable in [52]. The cardinality of a persistence diagram may carry important information in applications, especially for those homological features which die very quickly and may be considered as insignificant in the Wasserstain distance. Thus, the d_p^c distance accounts uneven cardinalities between persistence diagrams by assigning a regularization term with a parameter c, rather than connecting extra points to the diagonal. An example of different ways in matching is shown in Fig. 4.2. The d_p^c distance is defined as follows,

Definition 4.2. *Let D_X and D_Y be two persistence diagrams with cardinalities n and m respectively such that $n \leq m$ and denote $D_x = \{x_1, ..., x_n\}$, $D_y = \{y_1, ..., y_m\}$. Let $c > 0$ and $1 \leq p < \infty$ be fixed parameters. The d_p^c distance between two persistence diagrams D_x and D_y is*

$$d_p^c(D_x, D_y) = \left(\frac{1}{m} \left(\min_{\pi \in \Pi_m} \sum_{l=1}^{n} min(c, ||x_l - y_{\pi(l)}||_\infty)^p + c^P |m - n| \right) \right)^{\frac{1}{p}}, \tag{4.3}$$

where Π_m is the set of permutations of $(1, ..., m)$. If $m < n$, define $d_p^c(D_x, D_y) := d_p^c(D_y, D_x)$.

Eq. (4.3) shows that the d_p^c distance calculates the distance of points in two persistence diagrams without simulating points on the diagonal, it adds a penalty term of the difference in cardinalities between the two sets of points as well. The parameter c in Eq. (4.3) is a constant controlling the weight of penalization to be added in the d_p^c distance. A greater value of c yields a larger penalization. We tend to evaluate c between 0 and 1 as these have been empirically found to be appropriate options in real-world applications [52]. Moreover, A pair of p and c can be selected by running cross-validation on the data set.

Since persistence diagrams can summarize homological features of multiple dimensions in one diagram, such as in Fig. 4.3, the persistence diagram is generated by a set of five data points, which has been used as an example in Sect. 2.2.2, and this persistence diagram contains both 0-dim features (connected components) with cardinality 5 and 1-dim

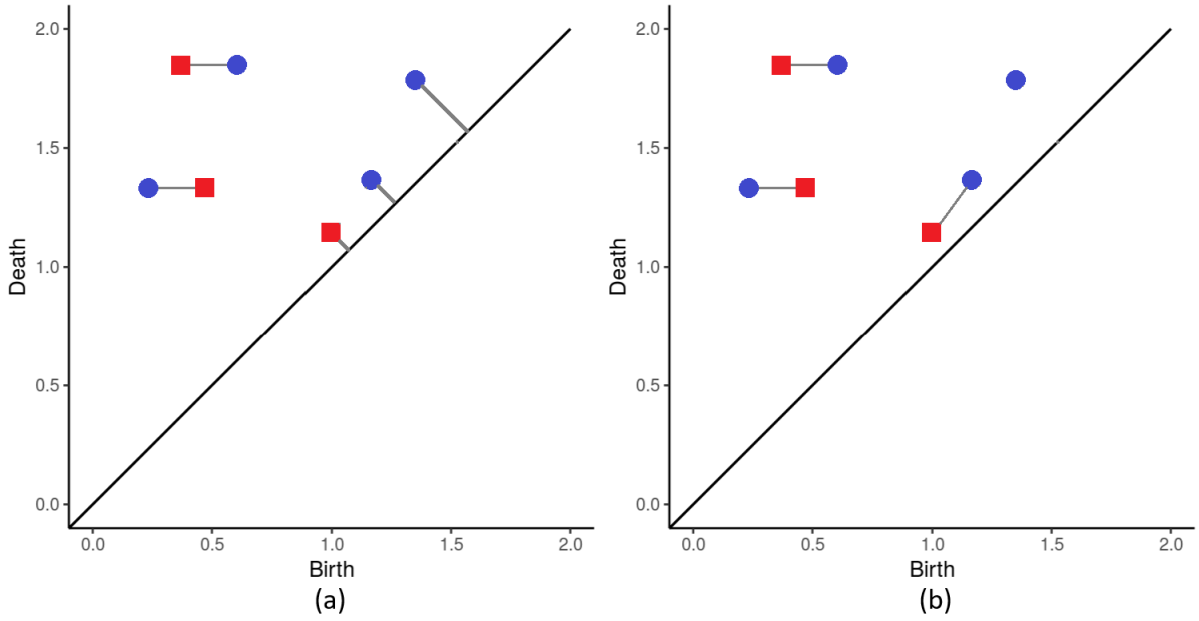

Figure 4.2: Given two persistence diagrams, one is represented by blue dots, the other one is represented by red squares. In panel (a), matching to diagonal is allowed in calculating Wasserstain distance, and the extra blue dot has been matched to the diagonal. In panel (b), the d_p^c distance does not match points to diagonal, and it counts the extra blue dot by adding a penalty term depending on parameter c as in Eq. (4.3).

features (holes) with cardinality 1, we consequently further define the d_p^c distance of a certain dimensional feature between a persistence diagram and a group of persistence diagrams.

Definition 4.3. *Consider only a specific β-dim homological feature, $\beta = 0, 1, 2, ...$, denote $\mathcal{C}$ as a collection of persistence diagrams from the same class, the d_p^c distance of β-dim homological feature between a persistence diagram D_x and a set of persistence diagrams $\mathcal{C}$ is given by its average,*

$$d_\beta(D_x, \mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{D \in \mathcal{C}} d_p^c(D_x, D), \tag{4.4}$$

where $|\mathcal{C}|$ represents the size of class $\mathcal{C}$.

Next, we build the d_p^c-based network classifier. For K classes of filament networks, every network in a class is generated under an identical set of cellular constraints. Therefore, we have K sets of persistence diagrams, where each set corresponds to a unique set of cellular

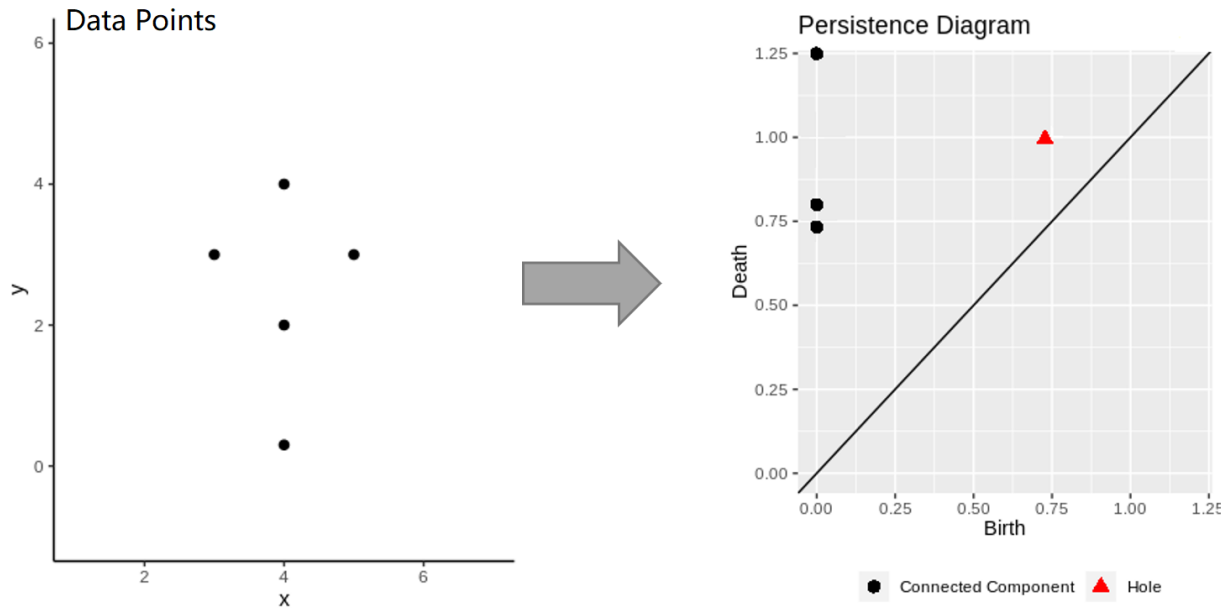

Figure 4.3: The persistence diagram on the left is generated by a set of five data points on the right. This persistence diagram contains both 0-dim features (connected components) with cardinality 5 and 1-dim features (holes) with cardinality 1. They have been used as an example in Sect. 2.2.2, the detailed generating process of the persistence diagram is exhibited in Fig. 2.3.

conditions. Given a new filament network with its persistence diagram D', our goal is to classify under which constraints the network was most likely generated, i.e. to which class k most likely belongs. We estimate this membership by calculating the distance between D' and each class of persistence diagrams. We then assign the new network to the class with the smallest distance. Additionally, we parameterize relative weights for different dimensions of homological feature in calculation of the distance and force the weights' sum to 1. The classifier is summarized in Algorithm 4.

4.3 Classification result

In our data set, we are provided with three classes of filament networks. Each class of filament networks is generated with a different number of cross-linker proteins (Class 1: 825 cross-linkers, Class 2: 1650 cross-linkers, Class 3: 3300 cross-linkers) in a cell bounded by

Algorithm 4 d_p^c-based network classifier

Let B is the highest dimension of homological features under consideration.
1. Take the training set $T_1, T_2, ..., T_K$ from each class of diagrams $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_K$,
2. For a new network with its corresponding persistence diagram D', compute

$$d(D', T_k) = \sum_{\beta=0}^{B} w_\beta d_\beta(D', T_k), \tag{4.5}$$

where $\sum_{\beta=0}^{B} w_\beta = 1$, and w_β determine how much β-dim homological feature is considered,
3. Assign D' a class label c' such that,

$$c' = \arg\min_{1 \le k \le K} d(D', T_k), \tag{4.6}$$

a 20 µm × 20 µm square. Each class contains a balanced number of 50 samples. Therefore, there are total of 150 individual filament networks.

In order to compare classifiers, we employ 10-fold Cross-Validation to estimate overall classification accuracy. All of the networks are randomly partitioned into 10 mutually exclusive sets, of which 9 partitions are selected as a training set, while the remaining 1 partition is used for testing. We repeat the classification procedure 10 times, such that every partition acts as a testing set exactly once. We consider the overall classification accuracy rate as the mean accuracy across all partitions. We also calculate the Area Under the ROC Curve (AUC) as our model performance indicator. AUC measures how much the classifier is capable of distinguishing between classes [58], ranging from 0 to 1. The higher the AUC and closer to 1, the better the classifier is at distinguishing networks from different classes.

4.3.1 Case I: Even weighted data analysis

In this case, we impose an additional restriction that each fold of filament network data is enforced to evenly have a same amount of filament networks from each class, when we randomly partition the data set into 10 folds. That is, each fold contains 5 random filament networks from the three classes respectively.

After making a persistence diagram based on the locations of actin beads for each filament network, we test our d_p^c-based classifier on this data set. Considering only 0-dim and 1-dim

homological features, we first run a sensitivity analysis on p and c in order to find the best pair of their values. We take p into account via a sequence of values $\{0.25, 0.5, 1, 2, 3, 4\}$, and grid search c in a sequence of values $\{0.1, 0.2, \cdots, 0.9, 1, 2, \cdots, 6\}$. In addition, we vary w_0, w_1 in Eq. (4.5), and record the best classification accuracy rate for every corresponding pair of p and c. The analysis results are plotted in Fig. 4.4,

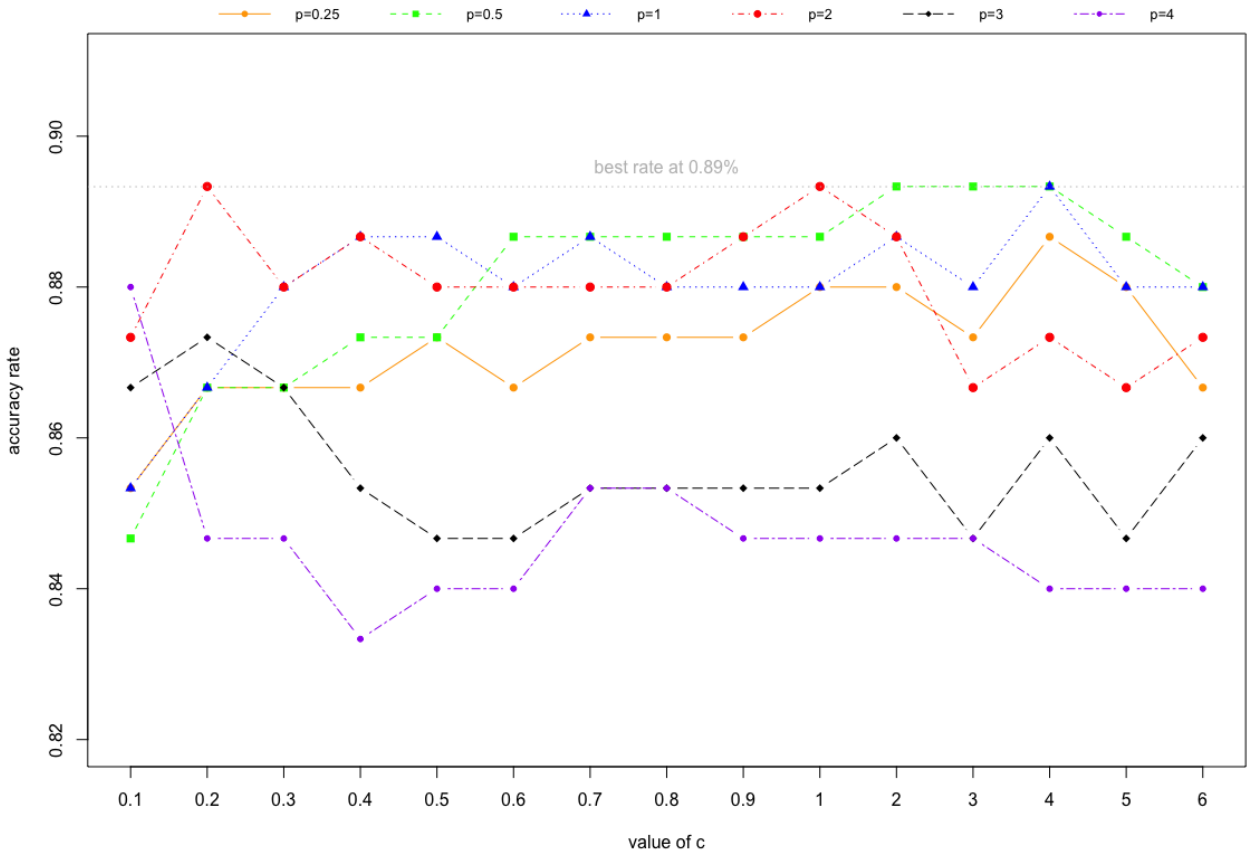

Figure 4.4: The curves in the figure show the classification accuracy rates corresponding to difference values of p and c. The d_p^c-based classifier achieves the overall best rate only when $p = 0.5$, $p = 1$, or $p = 2$.

Fig. 4.4 demonstrates that only when $p = 0.5$, $p = 1$, or $p = 2$, the d_p^c-based classifier achieves the overall best classification accuracy rate. Therefore, p should be chosen among these three values. Further more, when $p \leq 1$, choosing c around 4 provides a higher accuracy rate than any other values of c; when $p = 2$, higher accuracy rates are obtained at

both $c = 0.2$ and $c = 1$; when $p \geq 3$, a higher accuracy rate is obtained at $c = 0.2$ or even smaller c. Thus, these results also reveals the pattern that the larger value of p, the closer value to 0 of optimal c in the purpose to get a higher accuracy rate.

Here we chose $p = 2$ to mimic the tradition Euclidean distance. When $w_0 = 0.45, w_1 = 0.55$, which means connected components are considered slightly heavier than holes, and $c = 0.2$ to assign a smaller contribution from cardinality difference in the d_p^c distances, the best classification accuracy rate is 89%. The confusion matrix is displayed in Table 4.1. The AUC of our classifier is 0.94. Thus, using our d_p^c-based classifier methodology with 10-fold Cross-Validation technique, we classify 150 filament networks at a 89% accuracy rate, it also indicates our d_p^c-based classifier has an outstanding ability of distinguishing filament networks class by class.

Table 4.1: Confusion matrix

		True class		
		Class 1	Class 2	Class 3
Predicted	Class 1	50	3	0
	Class 2	0	40	6
	Class 3	0	7	44

We also compare our algorithm with other classifiers, which are built by properly vectorizing persistence diagrams, considering both 0-dim and 1-dim features and applying vector-based machine learning tools, such as Support Vector Machine (SVM) and Random Forest classifiers. Distance-statistics vectorization takes the similar idea in [52], vectorizing each persistence diagram by calculating statistics of the distances between itself and all other persistence diagrams. Precisely, for any two persistence diagrams D_i, D_j in a set of diagrams, let $d_\beta^{i,j}(D_i, D_j)$ be the distance of β-dim homological feature between D_i and D_j, further denote the mean and variance of the distances $\{d_\beta^{i,j}(D_i, D_j)\}_{j \neq i}$ by $\mathbb{E}_\beta^i$ and $\mathbb{V}_\beta^i$ respectively, then the persistence diagram D_i is vectorized as $(\mathbb{E}_0^i, \mathbb{V}_0^i, \mathbb{E}_1^i, \mathbb{V}_1^i)$. Moreover, we

could choose either the d_p^c or Wasserstein distance in computing $d_\beta^{i,j}(D_i, D_j)$. Persistence Image (PI) in [1] is a finite-dimensional vector representation of persistence diagrams. It is built by adding a Gaussian kernel density onto every point in the persistence diagram with a continuous weighting function, integrating the formed surface over a grid overlaid on this diagram. In addition, we compare our algorithm with a non-TDA-used classifier. A raster image is defined by a pixel that has one or more numbers associated with it. It has been wildly used in applications of image storage and geographic information systems. We place a 20×20 grid directly on the image of filament networks as shown in Fig. 4.1(b), and count the number of actin beads in every 1 μm^2 grid, then apply other common standard classification algorithms. All results of the accuracy rates and AUCs by using different classifiers are listed in Table 4.2.

Table 4.2: Accuracy rate and AUC for even weighted data

classifier	**accuracy**	**AUC**
d_p^c-based	89%	0.94
d_p^c-statistics SVM	86%	0.93
d_p^c-statistics Random Forest	84%	0.92
Wasserstein-based	83%	0.91
Wasserstein-statistics SVM	77%	0.86
Persistence Image SVM	75%	0.85
Wasserstein-statistics Random Forest	71%	0.79
Raster SVM	65%	0.77
Raster Random Forest	55%	0.69

From the results, the d_p^c distance generally outperforms the Wasserstein distance as the d_p^c distance adds a term to count difference cardinalities between persistence diagrams. All classifiers that use persistence diagrams perform better than the classifiers that do not use

TDA technique since persistence diagrams are embedded with extra geometric information hidden in the data, this information provides more clue in classification. But overall, the d_p^c-based classifier has the greatest accuracy classification rate and the highest AUC, it is superior than other classifiers.

4.3.2 Case II: Mixed data analysis

In this case, we remove the even weighted restriction as in Case I, when 10-folds Cross-Validation is performed. In this convention, the 150 filament networks are just randomly divided into 10 folds without any additional restriction, every fold is allowed to have unbalanced amounts of filament networks from each class, or even without filament networks from a class at all. Then let's investigate if our classifier is still robust when the training set is a set of mixed filament networks from the three classes.

We test our d_p^c-based classifier and compare with some classifiers that performed well in Case I, the results are exhibited in Table 4.3.

Table 4.3: Accuracy rate and AUC for mixed data

classifier	**accuracy**	**AUC**
d_p^c-based	88%	0.94
d_p^c-statistics SVM	87%	0.93
d_p^c-statistics Random Forest	85%	0.92
Wasserstein-based	83%	0.91
Wasserstein-statistics SVM	75%	0.83
Wasserstein-statistics Random Forest	73%	0.8
Persistence Image SVM	72%	0.79

Comparing to Table 4.2, the classification results in Table 4.3 are slightly different. This shows that our classifier with the 10-fold Cross-Validation technique handles mixed data very well. Moreover, the d_p^c-based classifier still does the best job in classification.

4.3.3 Case III: Weight analysis on filament networks

In this section, we vary the weights of different classes of filament networks in the sample set. We first look into the effect by over-weighting a certain class of filament networks. Since there are 50 networks in each class, we keep only one class with 50 networks and randomly remove 20 networks from the other two classes respectively. We test our d_p^c-based classifier with some other classifiers on these three cases, and show results in Table 4.4.

Table 4.4: Accuracy rate and AUC for weighted data

class weight	50 : 30 : 30		30 : 50 : 30		30 : 30 : 50	
classifier	**accuracy**	**AUC**	**accuracy**	**AUC**	**accuracy**	**AUC**
d_p^c-based	93%	0.96	86%	0.93	89%	0.95
d_p^c-statistics SVM	88%	0.91	83%	0.91	88%	0.94
Wasserstein-based	86%	0.92	79%	0.89	84%	0.92
d_p^c-statistics Random Forest	85%	0.91	84%	0.93	84%	0.91
Wasserstein-statistics Random Forest	75%	0.79	73%	0.79	75%	0.79
Wasserstein-statistics SVM	72%	0.78	72%	0.77	71%	0.78

From the results in Table 4.4, most classifiers indicate that heavier weight of Class 1 results in higher accuracy rate and AUC, while heavier weight of Class 2 results in lower accuracy rate and AUC. This phenomenon gives a preliminary conclusion that Class 1 networks are distinct from the other two classes, and conversely, Class 2 networks are less distinguishable from Class 1 and Class 3. This conclusion will be further enhanced and explained in the next two paragraphs.

To further investigate the effect by varying the weight of one class, we fix two classes with all their 100 networks, varying the size of the rest class from 10 to 50 with an increment of 10 networks, then record the accuracy rates and AUCs by applying our d_p^c-based classifier. The results are exhibited in Fig. 4.5.

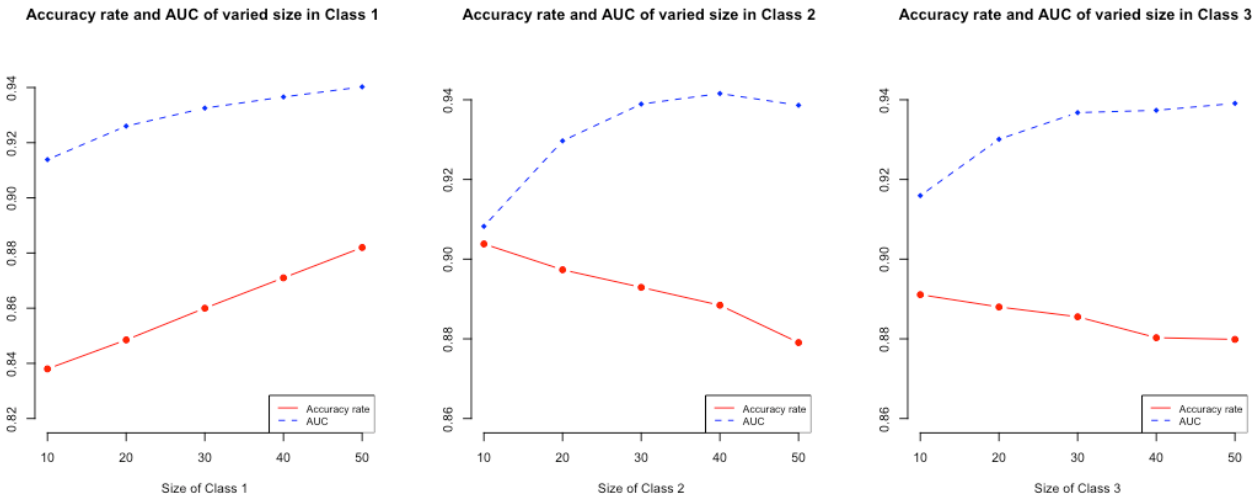

Figure 4.5: The three panels plot the accuracy rate curve and the AUC curve when the size is varied from 10 to 50 with an increment 10 in Class 1, Class 2 and Class 3 respectively.

Fig. 4.5 supports the conclusion that heavier weight of Class 1 results in higher accuracy rate and AUC, it also demonstrates that heavier weight of Class 2 or Class 3 will lower the accuracy rate since we have misclassifications in both Class 2 and Class 3. This is explained by consulting with the confusion matrix in Table 4.1. The first column of the confusion matrix reveals that we did perfect job in classifying Class 1 without any misclassification, thus the left panel of Fig. 4.5 shows both the accuracy rate and AUC increase significantly as the size of Class 1 is raised from 10 to 50. The second column of the confusion matrix indicates that we had 10 failures in classifying the networks in Class 2, including 3 misclassifications as Class 1 and 7 misclassifications as Class 3, so the increasing size of Class 2 will raise the probability of misclassifications, thus in the middle panel of Fig. 4.5, the accuracy rate decays as the size is varied from 10 to 50. But the AUC first increases as more reference

samples raise the capability of distinguishing networks for our model when we start from a relative small sample size 10, at some point the AUC levels off and then starts to decrease as more samples result in additional difficulties in distinguishing networks duo to larger variability. The third column of the confusion matrix demonstrates fewer misclassifications in Class 3, which leads to a gentle decay of the accuracy rate in the right penal of Fig. 4.5. The AUC also increases at first but with an almost flat growth rate when the weight of Class 3 further increases.

In summary, our proposed d_p^c-based classifier overwhelms other classifiers in classifying filament networks, it outperforms other classifiers with even weighted data and mixed data. In the classification results, the confusion matrix and the weight analysis demonstrate that the filament networks generated by 825 cross-linkers are highly distinguishable from the other two kinds of filament networks generated by 1650 and 3300 cross-linkers, i.e., 825 cross-linkers generate filament networks with unique features which can be uncovered by topological tools. But 1650 cross-linkers may generate few filament networks that carry close features as 825 or 3300 cross-linkers, while 3300 cross-linkers may also generate few networks that contain features as 1650 cross-linkers. Overall, our topological d_p^c distance-based classifier does great work in classifying these three classes of filament networks with high accuracy rate and AUC.

Chapter 5

Conclusions

In this dissertation, we have proposed a novel algorithm to track organelles in microscopy video. Our method combines Topological Data Analysis and advanced filtering techniques. A key features of our approach includes the adoption of topological data analysis principles, while using Gaussian processes and EnKF to facilitate the clustering involved in the computation of the associated nerve.

Unlike earlier tracking approaches, our method can proceed with, most importantly, or without a motion model. Without invoking a motion model, a reasonable guess may relax a strong requirement in the analysis of biological data, especially those obtained from in vivo microscopy at the level between cellular and molecular. In the opposite case, with a motion model, more precise inference can be incorporated into final estimation. In both cases, we estimate the displacement field from the data. In essence, our approach resembles data-driven clustering. However, our method implicitly assumes a phenomenological motion type that is exclusively informed by the observations. In any case, reconstructed tracks are valid if the computed of estimated displacement fields are consistent.

In our tracking method, the EnKF works in the case with minimal assumptions of the dynamic system. Moreover, it can achieve more precise results if additional knowledge of the dynamic system is provided. EnKF has the advantage that it inherits some trends from previous time step, and fits the nonlinear system, which could be more effectively applied when a sophisticated motion model is known in a real world analysis.

Earlier attempts obtained estimates of displacement fields using a combination of heuristics and conventional Kalman Filtering (KF). Given that motion patterns are non-linear, conventional KF lacks robustness and no longer works. Here, instead we develop a principled approach that models the displacement fields through Gaussian processes and we apply EnKF, which is shown to successfully estimate the desired dynamics under a wide range of motion conditions.

Overall, our intracellular tracking algorithm successfully reconstructs organelle trajectories as we show with example applications to synthetic and real data. This method optimizes parameters of organelles based on data captured in images, combine predictions with observations in estimating organelle movement, and link organelles based on topological analysis.

We then have proposed a d_p^c distance-based classifier on filament networks. This classifier enables us to successfully classify the filament networks, which are generated under different cellular environments. In this method, we combine a machine learning framework with topological data analysis. Our method is built on the foundation of persistent homology by encoding hidden topological features of the data into homological features. Key features of our method include summarizing homological features in persistence diagrams and adopting an advanced distance: the d_p^c distance. We test our classifier on the network data set with great success.

Our classification results show that the d_p^c classifier is superior than other classifiers, and TDA has been proved as a powerful tool in image data analysis again, yielding classification of actin filaments networks with high accuracy. Accordingly, this classification method could provides biologist with the opportunities to uncover the interaction of motor proteins, actin networks and streaming by comparing filament networks generated under different cellular environments. Our future work will clustering cells with real microscopy images. To that end, an unsupervised algorithm, which will be diagnostic to the number of different groups of data generated by the different number of cross-linkers, will be developed. The idea there is to establish a notion of mean or median for generationg a K-means or K-mediods type of clustering algorithm. Overall, our work is the first time to learn actin filament networks and

model it as a classification problem though TDA tools. Relying on these results, researchers could advance their understanding of cell physiology through this work.

Bibliography

[1] Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. (2017). Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, 18(1):218–252. 15, 43, 50

[2] Belchi, F., Pirashvili, M., Conway, J., Bennett, M., Djukanovic, R., and Brodzki, J. (2018). Lung topology characteristics in patients with chronic obstructive pulmonary disease. *Scientific reports*, 8(1):5341. 9

[3] Bendich, P., Chin, S. P., Clark, J., Desena, J., Harer, J., Munch, E., Newman, A., Porter, D., Rouse, D., Strawn, N., et al. (2016). Topological and statistical behavior classifiers for tracking applications. *IEEE Transactions on Aerospace and Electronic Systems*, 52(6):2644–2661. 3

[4] Bishop, C. M. (2006a). *Pattern recognition and machine learning*. springer.

[5] Bishop, G. (2006b). An introduction to the Kalman Filter. Technical report, TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, Monday. 5, 7, 26

[6] Blackman, S. S. (1986). Multiple-target tracking with radar applications. *Dedham, MA, Artech House, Inc., 1986, 463 p.* 3

[7] Bonis, T., Ovsjanikov, M., Oudot, S., and Chazal, F. (2016). Persistence-based pooling for shape pose recognition. In *International Workshop on Computational Topology in Image Context*, pages 19–29. Springer. 8

[8] Braun, S. S. v. and Schleiff, E. (2007). Movement of endosymbiotic organelles. *Current Protein and Peptide Science*, 8(5):426–438. 1

[9] Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, 16(1):77–102. 15, 43

[10] Cai, G., Parrotta, L., and Cresti, M. (2015). Organelle trafficking, the cytoskeleton, and pollen tube growth. *Journal of integrative plant biology*, 57(1):63–78. 1

[11] Cappé, O., Moulines, E., and Rydén, T. (2009). Inference in hidden Markov models. In *Proceedings of EUSFLAT conference*, pages 14–16.

[12] Carlsson, G. (2009). Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308. 3, 29

[13] Casella, G. and Berger, R. L. (2002). *Statistical inference*, volume 2. Duxbury Pacific Grove, CA. 18

[14] Chen, Y.-C. and Dobra, A. (2017). Measuring human activity spaces from gps data with density ranking and summary curves. *arXiv preprint arXiv:1708.05017*. 9

[15] Chen, Y.-C. et al. (2019). Generalized cluster trees and singular measures. *The Annals of Statistics*, 47(4):2174–2203. 9

[16] Chenouard, N., Bloch, I., and Olivo-Marin, J.-C. (2013). Multiple hypothesis tracking for cluttered biological image sequences. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2736–3750. 3

[17] Collings, D. A., Harper, J. D., Marc, J., Overall, R. L., and Mullen, R. T. (2002). Life in the fast lane: actin-based motility of plant peroxisomes. *Canadian Journal of Botany*, 80(4):430–441. 2

[18] Danuser, G. (2011). Computer vision in cell biology. *Cell*, 147(5):973–978. 2

[19] Derksen, J. (1996). Pollen tubes: a model system for plant cell growth. *Botanica Acta*, 109(5):341–345. 1

[20] Edelsbrunner, H. and Harer, J. (2010). *Computational topology: an introduction*. American Mathematical Soc. 3, 11, 29, 43

[21] Folland, G. B. (2013). *Real analysis: modern techniques and their applications*. John Wiley & Sons. 29

[22] Freedman, S. L., Banerjee, S., Hocky, G. M., and Dinner, A. R. (2017). A versatile framework for simulating the dynamic mechanical structure of cytoskeletal networks. *Biophysical journal*, 113(2):448–460. 1, 3, 41

[23] Freedman, S. L., Hocky, G. M., Banerjee, S., and Dinner, A. R. (2018). Nonequilibrium phase diagrams for actomyosin networks. *Soft matter*, 14(37):7740–7747. 3, 41

[24] Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC. 18

[25] Gutierrez, R., Lindeboom, J. J., Paredez, A. R., Emons, A. M. C., and Ehrhardt, D. W. (2009). Arabidopsis cortical microtubules position cellulose synthase delivery to the plasma membrane and interact with cellulose synthase trafficking compartments. *Nature cell biology*, 11(7):797. 2

[26] Hamada, T., Tominaga, M., Fukaya, T., Nakamura, M., Nakano, A., Watanabe, Y., Hashimoto, T., and Baskin, T. I. (2012). Rna processing bodies, peroxisomes, golgi bodies, mitochondria, and endoplasmic reticulum tubule junctions frequently pause at cortical microtubules. *Plant and Cell Physiology*, 53(4):699–708. 2

[27] Herman, B. and Albertini, D. F. (1984). A time-lapse video image intensification analysis of cytoplasmic organelle movements during endosome translocation. *The Journal of cell biology*, 98(2):565–576. 17

[28] Hiraoka, Y., Nakamura, T., Hirata, A., Escolar, E. G., Matsue, K., and Nishiura, Y. (2016). Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040. 8

[29] Hirsch, M., Wareham, R. J., Martin-Fernandez, M. L., Hobson, M. P., and Rolfe, D. J. (2013). A stochastic model for electron multiplication charge-coupled devices–from theory to practice. *PloS one*, 8(1):e53671. 17, 18

[30] Huang, B., Bates, M., and Zhuang, X. (2009). Super-resolution fluorescence microscopy. *Annual review of biochemistry*, 78:993–1016. 18

[31] Huang, F., Hartwich, T. M., Rivera-Molina, F. E., Lin, Y., Duim, W. C., Long, J. J., Uchil, P. D., Myers, J. R., Baird, M. A., Mothes, W., et al. (2013). Video-rate nanoscopy using sCMOS camera–specific single-molecule localization algorithms. *Nature methods*, 10(7):653. 18

[32] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

[33] Jaiswal, A., Godinez, W. J., Lehmann, M. J., and Rohr, K. (2016). Direct combination of multi-scale detection and multi-frame association for tracking of virus particles in microscopy image data. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 976–979. IEEE. 3

[34] Janesick, J. and Elliott, T. (1992). History and advancement of large array scientific ccd imagers. In *Astronomical CCD Observing and Reduction Techniques*, volume 23, page 1. 18

[35] Jazani, S., Sgouralis, I., and Pressé, S. (2019a). A method for single molecule tracking using a conventional single-focus confocal setup. *The Journal of chemical physics*, 150(11):114108.

[36] Jazani, S., Sgouralis, I., Shafraz, O. M., Levitus, M., Sivasankar, S., and Pressé, S. (2019b). An alternative framework for fluorescence correlation spectroscopy. *Nature communications*, 10.

[37] Kang, K., Maroulas, V., Schizas, I., and Bao, F. (2018). Improved distributed particle filters for tracking in a wireless sensor network. *Computational Statistics & Data Analysis*, 117:90–108. 3

[38] Kang, K., Maroulas, V., and Schizas, I. D. (2014). Drift homotopy particle filter for non-Gaussian multi-target tracking. In *17th International Conference on Information Fusion (FUSION)*, pages 1–7. IEEE.

[39] Kang, K., Maroulas, V., Schizas, I. D., and Blasch, E. (2016). A multilevel homotopy MCMC sequential Monte Carlo filter for multi-target tracking. In *Information Fusion (FUSION), 2016 19th International Conference on*, pages 2015–2021. IEEE. 3

[40] Kerber, M., Morozov, D., and Nigmetov, A. (2017). Geometry helps to compare persistence diagrams. *Journal of Experimental Algorithmics (JEA)*, 22:1–4. 43

[41] Law, K., Stuart, A., and Zygalakis, K. (2015). Data assimilation. *Cham, Switzerland: Springer*. 8, 24

[42] Lee, A., Tsekouras, K., Calderon, C., Bustamante, C., and Pressé, S. (2017). Unraveling the thousand word picture: an introduction to super-resolution data analysis. *Chemical reviews*, 117(11):7276–7330. 2, 3, 17, 18

[43] Li, C., Ovsjanikov, M., and Chazal, F. (2014). Persistence-based structural recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1995–2002. 8

[44] Lichtman, J. W. and Conchello, J.-A. (2005). Fluorescence microscopy. *Nature methods*, 2(12):910. 18

[45] Lichtscheidl, I. and Foissner, I. (1996). Video microscopy of dynamic plant cell organelles: principles of the technique and practical application. *Journal of Microscopy*, 181(2):117–128. 17, 19

[46] Liu, S., Mlodzianoski, M. J., Hu, Z., Ren, Y., McElmurry, K., Suter, D. M., and Huang, F. (2017). sCMOS noise-correction algorithm for microscopy images. *Nature methods*, 14(8):760. 18

[47] Lloyd, C. W. (1987). The plant cytoskeleton: the impact of fluorescence microscopy. *Annual Review of Plant Physiology*, 38(1):119–137. 17, 19

[48] Logan, D. C. and Leaver, C. J. (2000). Mitochondria-targeted gfp highlights the heterogeneity of mitochondrial shape, size and movement within living plant cells. *Journal of Experimental Botany*, 51(346):865–871. 2

[49] Mahler, R. P. and Maroulas, V. (2013). Tracking spawning objects. *IET Radar, Sonar & Navigation*, 7(3):321–331.

[50] Marchese, A. and Maroulas, V. (2016). Topological learning for acoustic signal identification. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1377–1381. IEEE. 8

[51] Marchese, A. and Maroulas, V. (2018). Signal classification with a point process distance on the space of persistence diagrams. *Advances in Data Analysis and Classification*, 12(3):657–682. 15, 44

[52] Maroulas, V., Micucci, C. P., and Spannaus, A. (2019a). A stable cardinality distance for topological classification. *Advances in Data Analysis and Classification*, pages 1–18. 15, 44, 49

[53] Maroulas, V., Mike, J. L., and Oballe, C. (2019b). Nonparametric estimation of probability density functions of random persistence diagrams. *Journal of Machine Learning Research*, 20(151):1–49. 13

[54] Maroulas, V., Nasrin, F., and Oballe, C. (2020). A bayesian framework for persistent homology. *SIAM Journal on Mathematics of Data Science*, 2(1):48–74. 8

[55] Maroulas, V. and Nebenführ (2015). Tracking rapid intracellular movements: a Bayesian random set approach. *The Annals of Applied Statistics*, 9(2):926–949. 1, 3

[56] Maroulas, V. and Stinis, P. (2012). Improved particle filters for multi-target tracking. *Journal of Computational Physics*, 231(2):602–611. 3

[57] Munkres, J. (2014). *Topology*. Pearson Education. 29

[58] Narkhede, S. (2018). Understanding auc-roc curve. *Towards Data Science*, 26. 47

[59] Nebenführ, A. (2014). Identifying subcellular protein localization with fluorescent protein fusions after transient expression in onion epidermal cells. In *Plant Cell Morphogenesis*, pages 77–85. Springer. 2

[60] Nebenführ, A. and Dixit, R. (2018). Kinesins and myosins: Molecular motors that coordinate cellular functions in plants. *Annual review of plant biology*, 69:329–361. 20

[61] Nebenführ, A., Gallagher, L. A., Dunahay, T. G., Frohlick, J. A., Mazurkiewicz, A. M., Meehl, J. B., and Staehelin, L. A. (1999). Stop-and-go movements of plant golgi stacks are mediated by the acto-myosin system. *Plant physiology*, 121(4):1127–1141. 2

[62] Nelson, B. K., Cai, X., and Nebenführ, A. (2007). A multicolored set of in vivo organelle markers for co-localization studies in arabidopsis and other plants. *The Plant Journal*, 51(6):1126–1136. 2

[63] Nicolau, M., Levine, A. J., and Carlsson, G. (2011). Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270. 9

[64] Pollard, T. D. and Cooper, J. A. (2009). Actin, a central player in cell shape and movement. *Science*, 326(5957):1208–1212. 1

[65] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[66] Raffel, M., Willert, C. E., Scarano, F., Kähler, C. J., Wereley, S. T., and Kompenhans, J. (2018). *Particle image velocimetry: a practical guide*. Springer.

[67] Ren, G., Maroulas, V., and Schizas, I. (2015). Distributed spatio-temporal association and tracking of multiple targets using multiple sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 51(4):2570–2589. 3

[68] Ren, G., Maroulas, V., and Schizas, I. D. (2016a). Decentralized sparsity-based multi-source association and state tracking. *Signal Processing*, 120:627–643.

[69] Ren, G., Maroulas, V., and Schizas, I. D. (2016b). Exploiting sensor mobility and covariance sparsity for distributed tracking of multiple sparse targets. *EURASIP Journal on Advances in Signal Processing*, 2016(1):53. 3

[70] Ren, G., Schizas, I. D., and Maroulas, V. (2014). Joint sensors-sources association and tracking. In *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 205–208. IEEE.

[71] Ren, G., Schizas, I. D., and Maroulas, V. (2016c). Sparsity based multi-target tracking using mobile sensors. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4578–4582. IEEE.

[72] Sbalzarini, I. F. and Koumoutsakos, P. (2005). Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of structural biology*, 151(2):182–195. 18

[73] Schizas, I. D. and Maroulas, V. (2015). Dynamic data driven sensor network selection and tracking. *Procedia Computer Science*, 51:2583–2592.

[Schizas et al.] Schizas, I. D., Maroulas, V., and Ren, G. Regularized kernel matrix decomposition for thermal video multi-object detection and tracking.

[75] Sgouralis, I., Nebenführ, A., and Maroulas, V. (2017). A Bayesian topological framework for the identification and reconstruction of subcellular motion. *SIAM Journal on Imaging Sciences*, 10(2):871–899. 3, 18, 26

[76] Shotton, D. M. (1988). Video-enhanced light microscopy and its applications in cell biology. *J Cell Sci*, 89(2):129–150. 17

[77] Singh, G., Mémoli, F., and Carlsson, G. E. (2007). Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *SPBG*, pages 91–100. 3, 29

[78] Singh, N., Couture, H. D., Marron, J., Perou, C., and Niethammer, M. (2014). Topological descriptors of histology images. In *International Workshop on Machine Learning in Medical Imaging*, pages 231–239. Springer. 9

[79] Smal, I., Draegestein, K., Galjart, N., Niessen, W., and Meijering, E. (2008). Particle filtering for multiple object tracking in dynamic fluorescence microscopy images: Application to microtubule growth analysis. *IEEE Transactions on Medical Imaging*, 27(6):789–804. 3

[80] Smal, I., Niessen, W., and Meijering, E. (2006). Particle filtering for multiple object tracking in molecular cell biology. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 129–132. IEEE. 3

[81] Snyder, D. L., Hammoud, A. M., and White, R. L. (1993). Image recovery from data acquired with a charge-coupled-device camera. *JOSA A*, 10(5):1014–1023. 17, 18

[82] Sousbie, T., Pichon, C., and Kawahara, H. (2011). The persistent cosmic web and its filamentary structure–ii. illustrations. *Monthly Notices of the Royal Astronomical Society*, 414(1):384–403. 9

[83] Spanier, E. H. (1989). *Algebraic topology*, volume 55. Springer Science & Business Media. 29

[84] Sparkes, I. A. (2010). Motoring around the plant cell: insights from plant myosins. 1

[85] Stone, L. D., Streit, R. L., Corwin, T. L., and Bell, K. L. (2013). *Bayesian multiple target tracking*. Artech House. 3

[86] Thomas, C., Tholl, S., Moes, D., Dieterle, M., Papuga, J., Moreau, F., and Steinmetz, A. (2009). Actin bundling in plants. *Cell motility and the cytoskeleton*, 66(11):940–957. 1

[87] van de Weygaert, R., Platen, E., Vegter, G., Eldering, B., and Kruithof, N. (2010). Alpha shape topology of the cosmic web. In *2010 International Symposium on Voronoi Diagrams in Science and Engineering*, pages 224–234. IEEE. 9

[88] Vick, J. K. and Nebenführ, A. (2012). Putting on the breaks: Regulating organelle movements in plant cells f. *Journal of integrative plant biology*, 54(11):868–874. 2, 41

[89] Wasserman, L. (2018). Topological data analysis. *Annual Review of Statistics and Its Application*, 5:501–532. 9, 43

[90] Willert, C. E. and Gharib, M. (1991). Digital particle image velocimetry. *Experiments in fluids*, 10(4):181–193.

[91] Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT Press Cambridge, MA.

[92] Zitova, B. and Flusser, J. (2003). Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000.

Appendices

A KF equations of the discretized displacement fields

A.1 Forward fields: y-components

$$\left[\begin{array}{c}\psi_{1,+}(\bar{\bar{x}}^1)\\ \vdots\\ \psi_{1,+}(\bar{\bar{x}}^\Lambda)\\ \hline \psi_{1,+}(\bar{x}^1)\\ \vdots\\ \psi_{1,+}(\bar{x}^J)\end{array}\right] \sim N_{\Lambda+J}\left(\left[\begin{array}{c}0\\ \vdots\\ 0\\ \hline 0\\ \vdots\\ 0\end{array}\right], \left[\begin{array}{ccc|ccc} K(\bar{\bar{x}}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^1,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{x}^J)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^\Lambda,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{x}^J)\\ \hline K(\bar{x}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^1,\bar{\bar{x}}^\Lambda) & K(\bar{x}^1,\bar{x}^1) & \cdots & K(\bar{x}^1,\bar{x}^J)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ K(\bar{x}^J,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^J,\bar{\bar{x}}^\Lambda) & K(\bar{x}^J,\bar{x}^1) & \cdots & K(\bar{x}^J,\bar{x}^J) \end{array}\right]\right).$$

$$\left[\begin{array}{c}\psi_{n,+}(\bar{\bar{x}}^1)\\ \vdots\\ \psi_{n,+}(\bar{\bar{x}}^\Lambda)\\ \hline \psi_{n,+}(\bar{x}^1)\\ \vdots\\ \psi_{n,+}(\bar{x}^J)\end{array}\right] \sim N_{\Lambda+J}\left(\Psi_+\left(\left[\begin{array}{c}\psi_{n-1,+}(\bar{\bar{x}}^1)\\ \vdots\\ \psi_{n-1,+}(\bar{\bar{x}}^\Lambda)\\ \hline \psi_{n-1,+}(\bar{x}^1)\\ \vdots\\ \psi_{n-1,+}(\bar{x}^J)\end{array}\right]\right), \left[\begin{array}{ccc|ccc} K(\bar{\bar{x}}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^1,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{x}^J)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^\Lambda,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{x}^J)\\ \hline K(\bar{x}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^1,\bar{\bar{x}}^\Lambda) & K(\bar{x}^1,\bar{x}^1) & \cdots & K(\bar{x}^1,\bar{x}^J)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ K(\bar{x}^J,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^J,\bar{\bar{x}}^\Lambda) & K(\bar{x}^J,\bar{x}^1) & \cdots & K(\bar{x}^J,\bar{x}^J) \end{array}\right]\right), \qquad n = 2,\ldots,N-1.$$

$$\begin{bmatrix} \bar{\psi}^1_{n,+} \\ \vdots \\ \bar{\psi}^J_{n,+} \end{bmatrix} \sim N_J \left(\left[\begin{array}{ccc|ccc} 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{array}\right] \left[\begin{array}{c} \psi_{n,+}(\bar{\bar{x}}^1) \\ \vdots \\ \psi_{n,+}(\bar{\bar{x}}^\Lambda) \\ \hline \psi_{n,+}(\bar{x}^1) \\ \vdots \\ \psi_{n,+}(\bar{x}^J) \end{array}\right], \begin{bmatrix} \sigma_v^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_v^2 \end{bmatrix} \right), \qquad n = 1, \ldots, N-1.$$

A.2 Backward fields: x-components

$$\left[\begin{array}{c} \phi_{N,-}(\bar{\bar{x}}^1) \\ \vdots \\ \phi_{N,-}(\bar{\bar{x}}^\Lambda) \\ \hline \phi_{N,-}(\bar{x}^1) \\ \vdots \\ \phi_{N,-}(\bar{x}^J) \end{array}\right] \sim N_{\Lambda+J} \left(\left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \end{array}\right], \left[\begin{array}{ccc|ccc} K(\bar{\bar{x}}^1, \bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^1, \bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^1, \bar{x}^1) & \cdots & K(\bar{\bar{x}}^1, \bar{x}^J) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ K(\bar{\bar{x}}^\Lambda, \bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^\Lambda, \bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^\Lambda, \bar{x}^1) & \cdots & K(\bar{\bar{x}}^\Lambda, \bar{x}^J) \\ \hline K(\bar{x}^1, \bar{\bar{x}}^1) & \cdots & K(\bar{x}^1, \bar{\bar{x}}^\Lambda) & K(\bar{x}^1, \bar{x}^1) & \cdots & K(\bar{x}^1, \bar{x}^J) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ K(\bar{x}^J, \bar{\bar{x}}^1) & \cdots & K(\bar{x}^J, \bar{\bar{x}}^\Lambda) & K(\bar{x}^J, \bar{x}^1) & \cdots & K(\bar{x}^J, \bar{x}^J) \end{array}\right] \right)$$

$$\left[\begin{array}{c}\phi_{n,-}(\bar{\bar{x}}^1)\\ \vdots\\ \phi_{n,-}(\bar{\bar{x}}^\Lambda)\\ \hline \phi_{n,-}(\bar{x}^1)\\ \vdots\\ \phi_{n,-}(\bar{x}^J)\end{array}\right] \sim N_{\Lambda+J}\left(\Psi_-\left(\left[\begin{array}{c}\phi_{n+1,-}(\bar{\bar{x}}^1)\\ \vdots\\ \phi_{n+1,-}(\bar{\bar{x}}^\Lambda)\\ \hline \phi_{n+1,-}(\bar{x}^1)\\ \vdots\\ \phi_{n+1,-}(\bar{x}^J)\end{array}\right]\right), \left[\begin{array}{ccc|ccc} K(\bar{\bar{x}}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^1,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{x}^J)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^\Lambda,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{x}^J)\\ \hline K(\bar{x}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^1,\bar{\bar{x}}^\Lambda) & K(\bar{x}^1,\bar{x}^1) & \cdots & K(\bar{x}^1,\bar{x}^J)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ K(\bar{x}^J,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^J,\bar{\bar{x}}^\Lambda) & K(\bar{x}^J,\bar{x}^1) & \cdots & K(\bar{x}^J,\bar{x}^J)\end{array}\right]\right), \qquad n=2,\ldots,N-1.$$

$$\left[\begin{array}{c}\bar{\phi}^1_{n,-}\\ \vdots\\ \bar{\phi}^J_{n,-}\end{array}\right] \sim N_J\left(\left[\begin{array}{ccc|ccc} 0 & \cdots & 0 & 1 & \cdots & 0\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ 0 & \cdots & 0 & 0 & \cdots & 1\end{array}\right]\left[\begin{array}{c}\phi_{n,-}(\bar{\bar{x}}^1)\\ \vdots\\ \phi_{n,-}(\bar{\bar{x}}^\Lambda)\\ \hline \phi_{n,-}(\bar{x}^1)\\ \vdots\\ \phi_{n,-}(\bar{x}^J)\end{array}\right], \left[\begin{array}{ccc}\sigma_v^2 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \sigma_v^2\end{array}\right]\right), \qquad n=2,\ldots,N.$$

A.3 Backward fields: y-components

$$\left[\begin{array}{c}\psi_{N,-}(\bar{\bar{x}}^1)\\ \vdots\\ \psi_{N,-}(\bar{\bar{x}}^\Lambda)\\ \hline \psi_{N,-}(\bar{x}^1)\\ \vdots\\ \psi_{N,-}(\bar{x}^J)\end{array}\right] \sim N_{\Lambda+J}\left(\left[\begin{array}{c}0\\ \vdots\\ 0\\ \hline 0\\ \vdots\\ 0\end{array}\right], \left[\begin{array}{ccc|ccc} K(\bar{\bar{x}}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^1,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{x}^J)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^\Lambda,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{x}^J)\\ \hline K(\bar{x}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^1,\bar{\bar{x}}^\Lambda) & K(\bar{x}^1,\bar{x}^1) & \cdots & K(\bar{x}^1,\bar{x}^J)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ K(\bar{x}^J,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^J,\bar{\bar{x}}^\Lambda) & K(\bar{x}^J,\bar{x}^1) & \cdots & K(\bar{x}^J,\bar{x}^J)\end{array}\right]\right)$$

$$\left[\begin{array}{c}\psi_{n,-}(\bar{\bar{x}}^1)\\ \vdots\\ \psi_{n,-}(\bar{\bar{x}}^\Lambda)\\ \hline \psi_{n,-}(\bar{x}^1)\\ \vdots\\ \psi_{n,-}(\bar{x}^J)\end{array}\right] \sim N_{\Lambda+J}\left(\Psi_-\left(\left[\begin{array}{c}\psi_{n+1,-}(\bar{\bar{x}}^1)\\ \vdots\\ \psi_{n+1,-}(\bar{\bar{x}}^\Lambda)\\ \hline \psi_{n+1,-}(\bar{x}^1)\\ \vdots\\ \psi_{n+1,-}(\bar{x}^J)\end{array}\right]\right), \left[\begin{array}{ccc|ccc} K(\bar{\bar{x}}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^1,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^1,\bar{x}^J)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{\bar{x}}^\Lambda) & K(\bar{\bar{x}}^\Lambda,\bar{x}^1) & \cdots & K(\bar{\bar{x}}^\Lambda,\bar{x}^J)\\ \hline K(\bar{x}^1,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^1,\bar{\bar{x}}^\Lambda) & K(\bar{x}^1,\bar{x}^1) & \cdots & K(\bar{x}^1,\bar{x}^J)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ K(\bar{x}^J,\bar{\bar{x}}^1) & \cdots & K(\bar{x}^J,\bar{\bar{x}}^\Lambda) & K(\bar{x}^J,\bar{x}^1) & \cdots & K(\bar{x}^J,\bar{x}^J)\end{array}\right]\right), \qquad n=2,\dots,N-1.$$

$$\begin{bmatrix} \bar{\psi}^1_{n,-} \\ \vdots \\ \bar{\psi}^J_{n,-} \end{bmatrix} \sim N_J \left(\left[\begin{array}{ccc|ccc} 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{array}\right] \left[\begin{array}{c} \psi_{n,-}(\bar{\bar{x}}^1) \\ \vdots \\ \psi_{n,-}(\bar{\bar{x}}^\Lambda) \\ \hline \psi_{n,-}(\bar{x}^1) \\ \vdots \\ \psi_{n,-}(\bar{x}^J) \end{array}\right], \begin{bmatrix} \sigma_v^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_v^2 \end{bmatrix} \right), \qquad n = 2, \ldots, N.$$

Vita

Le Yin was born in Zhejiang province, China. He attended Ningbo University to study applied mathematics in 2007 and received a Bachelor of Science degree three years later. He was then transferred to Middle Tennessee State University in 2010, he got another Bachelor of Science degree in 2012 and graduated with a Master of Science degree under the supervision of Prof. Don Hong in 2014.

In August 2014, he joined the Department of Mathematics at the University of Tennessee, Knoxville as a graduate student, concentrating on Statistics and Data Analysis.