DATA SCIENCE METHODS FOR ANALYZING NANOMATERIAL IMAGES AND VIDEOS

A Dissertation

by

YANJUN QIAN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Yu Ding
Co-Chair of Committee,	Jianhua Huang
Committee Members,	Satish Bukkapatnam
	Hong Liang
Head of Department,	Mark Lawley

August 2018

Major Subject: Industrial Engineering

ABSTRACT

A large amount of nanomaterial characterization data has been routinely collected by using electron microscopes and stored in image or video formats. A bottleneck in making effective use of the image/video data is the lack of the development of sophisticated data science methods capable of unlocking valuable material pertinent information buried in the raw data. To address this problem, the research of this dissertation begins with understanding the physical mechanisms behind the concerned process to determine why the generic methods fall short. Afterwards, it designs and improves image processing and statistical modeling tools to address the practical challenges. Specifically, this dissertation consists of two main tasks: extracting useful information from images or videos of nanomaterials captured by electron microscopes, and designing analytical methods for modeling/monitoring the dynamic growth of nanoparticles. In the first task, a two-pipeline framework is proposed to fuse two kinds of image information for nanoscale object detection that can accurately identify and measure nanoparticles in transmission electron microscope (TEM) images of high noise and low contrast. To handle the second task of analyzing nanoparticle growth, this dissertation develops dynamic nonparametric models for time-varying probability density functions (PDFs) estimation. Unlike simple statistics, a PDF contains fuller information about the nanoscale objects of interests. Characterizing the dynamic changes of the PDF as the nanoparticles grow into different sizes and morph into different shapes, the proposed nonparametric methods are capable of analyzing an *in situ* TEM video to delineate growth stages in a retrospective analysis, or tracking the nanoparticle growth process in a prospective analysis. The resulting analytic methods have applications in areas beyond the nanoparticle growth process such as the image-based process control tasks in additive manufacturing.

DEDICATION

I dedicate this dissertation to my family, who are always caring and supporting me during my Ph.D. study.

This dissertation is dedicated to my mother, Suifang Liu, my father, Guoping Qian and my grandmother, Ruizhen Zhu, who give me unconditional love and understanding in spite of the distance between us.

This dissertation is also dedicated to my wife, Ya Su, for accompanying me throughout this long journey with her patience and kindness and being a constant source of encouragement and help in my life.

ACKNOWLEDGMENTS

I would like to express my gratitude and appreciation to my advisor, Prof. Yu Ding, without whom this work would have never been accomplished. He provided me the opportunity to join his lab, and since then he always trusted my research ability and provided support and guidance in the face of many obstacles. I am indebted to him for the academic training he gave me, such as cultivating critical thinking, building confidence to overcome challenges and improving professional communication skills, to name a few. His personality inspires me to become a researcher and mentor like him in the future.

I would also thank my co-advisor, Prof. Jianhua Huang, for guiding me to statistical science. He helped me accumulate the knowledge of statistics systematically and provided numerous invaluable suggestions to my research. His experience on non-parametric modeling was of vital importance to this work. I am grateful to the rest of my committee members, Prof. Satish Bukkapatnam and Prof. Hong Liang, who gave me a lot of insightful feedback on material and manufacturing, thus broadening my research scope during this dissertation study.

Finally, I would like to thank Prof. Chiwoo Park, whose research laid the foundation of my works in this dissertation. During years of collaboration, he became a role model for me to pursue academic achievements. I am also thankful to my labmates, namely Arash Pourhabib, Shilan Jin, Hoon Hwangbo, Erika Sy, Ahmed Aziz Ezzat, Imtiaz Ahmed, Jason Lawley, David Pérez, Qian Wu, and Doowon Choi, for creating a friendly and collaborative working environment.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Yu Ding and Satish Bukkapatnam of the Department of Industrial & Systems Engineering, Professor Jianhua Huang of the Department of Statistics and Professor Hong Liang of the Department of Mechanical Engineering.

The electron microscopic images analyzed in Chapter 2 were provided by Dr. Xiaodong Li in the Academy of Mathematics and Systems Science of Chinese Academy of Sciences.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by a Graduate Teaching Fellowship from Texas A&M University and research grants FA9550-13-1-0075 and FA9550-17-1-0044 from AFOSR Dynamic Data-Driven Application System program.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Motivation

Thanks to fast-developing nanotechnology, advanced manufacturing can produce materials with special mechanical, electrical, and optical properties by mixing nanoscale particles into the host materials. The recent progress in dynamic nanoscale imaging technologies, e.g., the introduction of *in situ* transmission electron microscope (TEM) imaging [1], provides powerful tools to analyze and monitor the nanomanufacturing processes. By taking dynamic nanoscale images, *in situ* TEM enables us to observe the nanoparticle growth pathways directly and provides an unprecedented opportunity for material scientists to look closely into the nanoscale world.

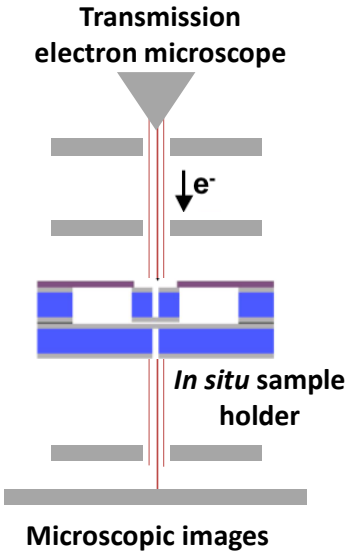


Figure 1.1: The structure of an *in situ* TEM.

Figure 1.1 illustrates the basic structure of an *in situ* TEM. Material scientists put a growth solution into an *in situ* sample holder that is about 200nm in thickness. After the growth is initialized, the TEM will capture the process in the solution by taking consecutive images. The current *in situ* TEMs can take 15 image frames per second and the resolution of each frame can run up to 1000×1000 pixels.

In this dissertation, our research focuses on the nanoparticle self-assembly process [3, 4], illustrated in Figure 1.2. Initialized by an electron beam, small building blocks, like atoms, ions and molecules, form into cores simultaneously in a solution. Under certain conditions, the cores continue to grow into particles with an ordered structure. When the process is completed, nano-super-lattices [5] with unique properties are obtained. The self-assembly process is considered as one of the promising bottom-up methods for the large-scale nanomanufacturing process. However, due to the randomness in the process, the self-assembled nanoparticles tend to have a size/shape distribution with large variance. A control strategy should be applied to the self-assembly process to ensure a concentrated size/shape distribution. To enable this type of process control, it becomes almost a prerequisite to first develop a data-driven, dynamic model that can track and anticipate nanoparticle growth.

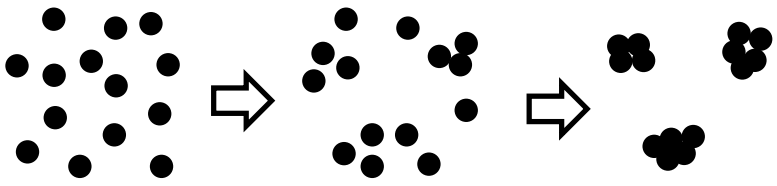


Figure 1.2: Illustration of the nanoparticle self-assembly growth process.

To develop such a dynamic model, our research objective entails three tasks: (1) a

robust processing of static TEM images that have poor image quality, (2) a retrospective analysis of *in situ* TEM videos, and (3) a prospective analysis of *in situ* TEM videos. We begin with an algorithm that can detect nanoparticles from a noisy background in low-quality TEM images for measuring their properties. This work lays the basis for the two subsequent tasks, as the individual frames of *in situ* TEM videos are usually images with heavy noise. Centering on off-line TEM videos, our second task is to propose a retrospective analyzing framework. With the estimated growth trajectory from an entire TEM video, we can signal possible stage changes in the process and delineate the stages of growth, providing valuable information to aid material scientists in their discoveries. Our third task is a prospective analysis for *in situ* TEM videos. We develop a dynamic and forward-looking model that can track the nanoparticles' growth trajectory as they progress. Our dynamic model functions like a Kalman filter in modern control systems, helping us identify the necessary adjustments and control actions to produce nanoparticles of desired shapes and/or sizes. The interrelationship of the three tasks is shown in Figure 1.3.

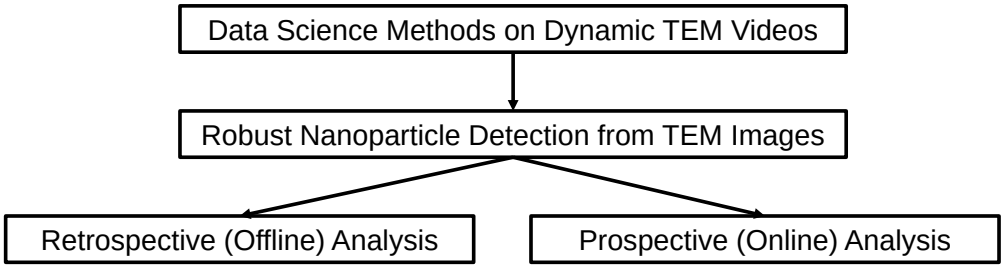


Figure 1.3: The interrelationship of the proposed methods for dynamic nanoimaging data.

1.2 Static and Dynamic TEM Image Processing

1.2.1 Static TEM Image Processing

To analyze *in situ* TEM videos, the first step is to detect and measure nanoparticles from each image frame. While there is plenty of research on object detection in computer science, not many dedicated methods for nanoparticle detection have been developed. One of the commonly used tools for analyzing nanoimages is `ImageJ` [6], a freeware developed by the National Institutes of Health (NIH). However, this tool was devised for bioimage processing to detect cells and neurons from microscopic images. When we apply `ImageJ` to nanoimages, the results of nanoparticle detection are usually not satisfying. The major challenge encountered is segmenting the overlapping particles. Unlike cells, nanoparticles do not have observable nuclei, the presence of which can give strong clues for separating two cells. Without such clues, it becomes much more difficult to segment the overlapping nanoparticles and infer their contours and shapes.

To overcome this challenge, specialized algorithms have been developed to detect nanoparticles from TEM images and identify their shapes. Park et al. [7] proposed a three-stage approach to overcome the challenge. First, it separates overlapping particles based on the criterion that nanoparticles tend to have convex shapes. Then with functional principle component analysis (PCA) [8], the whole contours of particles are recovered, and their shapes are classified via a k-nearest neighbors (K-NN) method. Park et al. [9] proposed to combine the contour recovery and shape classification by a learning approach. The contours are represented by a Gaussian mixture model of B-splines, and the missing contours and shape classes are jointly learned by the expectation-conditional maximization (ECM) algorithm [10]. Furthermore, Konomi et al. [11] provided a Bayesian framework for shape analysis. A dictionary of the predetermined shape families is learned from the training TEM images. Then a marked-point process [12] is established to represent the

particles, where the markers determine the shape information and the points indicate the locations. The process can be inferred through a Markov chain Monte Carlo (MCMC) method, and the morphology characteristics of nanoparticles are classified automatically.

The methods discussed above have shown good performance when applied to TEM images with relative low noise and high quality. When we apply them to images of high noise and low contrast, which are common for images in *in situ* TEM videos, the detection results are unsatisfactory. Because nanoparticle detection is the foundation of further video analysis, it is critical to have a detecting algorithm with both robustness and accuracy. In the first task of this dissertation, we will focus on detecting nanoparticles from noisy and low-contrast TEM images.

1.2.2 Dynamic TEM Image Processing

After we identify and measure nanoparticles from image frames of *in situ* TEM videos, the next step is to model and analyze the dynamic growth process. In computer science, video monitoring is usually accomplished by multiple object tracking. The motion of each individual object is tracked through the video, then the characteristics and interactions of all the objects are summarized to model the dynamic process. The method based on object tracking is widely used to monitor vehicles and people in traffic and surveillance videos [13, 14].

Researchers have proposed dynamic methods to model an individual particle's growth pathway observed in TEM videos. Park et al. [15] proposed a Bayesian algorithm to characterize the growth pathway with non-longitudinal TEM images. The contours of nanoparticles are represented by a Gaussian mixture model of B-splines with a non-decreasing constraint, and then their multiple possible pathways are modeled by a Dirichlet process. After the parameters of the Bayesian model are learned from TEM images captured at different times, the major growth trajectories of nanoparticles through the growth process are

identified automatically. Park et al. [16] proposed another method to track the interactions of different nanoparticles. After the particles on each image frame are detected, all the possible interactions between neighboring particles are formulated as an energy function. By optimizing the energy function through all the video frames, Park et al. [16] detected the interactions that occurred in the process and track the growth pathway of any individual nanoparticle.

However, when the quality of TEM videos is low, accurately tracking each nanoparticle becomes difficult, if not impossible. Unlike people or vehicles in surveillance videos, nanoparticles lack trackable features. Moreover, tracking every single particle may not be necessary. In fact, material scientists care more about the collective behaviors of nanoparticles, reflected more fully in the dynamic probability density function (PDF) of particles' sizes that evolves in a particle growth process. To model the dynamic particle size distribution, we develop both offline and online methods to learn the time-varying PDF from the nanoparticle images captured in *in situ* TEM videos.

1.2.3 Statistical Methods for Learning Dynamic Distribution

In statistics research, there are two schools of thought to characterize a PDF from observations: parametric methods and non-parametric ones. As no parametric model can fit all possible PDFs of an unknown growth process, we will focus on those non-parametric methods in this dissertation. The kernel estimator [17] and penalized B-splines [18] are two popular non-parametric density estimation methods. The kernel estimator calculates a summation of local kernels around all the observations to obtain the PDF, while B-splines fit a smooth curve represented by a linear combination of basis from the observed histogram. Both methods show strong capabilities to estimate a static PDF, but it is still a challenging problem to extend them for a dynamic PDF estimation.

Recent works [19, 20, 21, 22] proposed a state-space method to model a time-varying

PDF, shown in Figure 1.4. A hidden variable $\boldsymbol{\alpha}_t$ changes through time, indicating the underlying dynamics of the process, and a PDF $F_{\boldsymbol{\alpha}_t}(r)$ will change accordingly. At each time, we will observe r_{it} following the corresponding $F_{\boldsymbol{\alpha}_t}(r)$. To implement the framework, we would like to establish a non-parametric model between $\boldsymbol{\alpha}_t$ and $F_{\boldsymbol{\alpha}_t}(r)$, and estimate them efficiently from the observed r_{it}.

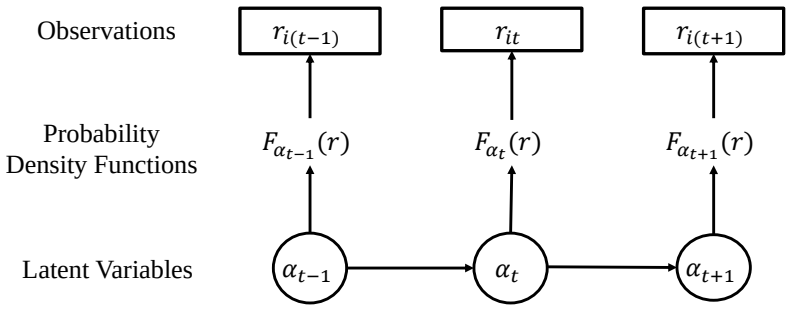


Figure 1.4: The state-space model for esimating a time-varying distribution function.

The existing methods adopt different approaches to address these two tasks. In [20, 22], $F_{\boldsymbol{\alpha}_t}(r)$ is represented by a kernel estimator while $\boldsymbol{\alpha}_t$ is the weights of the local kernels. The hidden states and the distributions are then estimated by sampling their posterior distributions via MCMC. In [19, 21], $F_{\boldsymbol{\alpha}_t}(r)$ is approximated by the observed histograms while $\boldsymbol{\alpha}_t$ indicates their underlying means. A particle filter [23] is devised to estimate $\boldsymbol{\alpha}_t$ from the observations. However, neither of the existing approaches meet the online requirements for a prospective analysis. Both methods estimate the distribution by a sampling algorithm, which tends to have high computational complexity when the dimension $\boldsymbol{\alpha}_t$ becomes large. In the second and third tasks of this dissertation, we develop both retrospective and prospective methods based on a B-spline model [18]. One advantage of our proposed approach is that the resulting methods can estimate the time-varying PDF of

particle size efficiently and model the dynamics of self-assembly growth effectively.

As this point it is appropriate to highlight the difference between the retrospective and prospective model. In the retrospective analysis, the goal is to estimate the time-varying distribution from all the data captured in a finished process, whereas in the prospective analysis, we focus on an ongoing process and monitor the growth status by estimating the evolving particle size distribution. Speed is very essential for the prospective analysis: the estimation should be fast enough to catch up with the video's updating that is as fast as 15 frames per second for the video data we have at hand. By contrast, while a fast method is always desired, processing speed is not essential in the retrospective analysis.

1.3 Structure of This Dissertation

The structure of the following chapters is as follows. In Chapter 2, we present a nanoparticle detection method for TEM images of heavy noise and low contrast. To improve the accuracy and robustness of the detection, we propose a two-pipeline framework, in which one pipeline mainly uses intensity-based information for detection, while the other one focuses on gradient-based information. Then, we formulate a binary integer programming (BIP) problem to select particles from two sets of results detected by the two pipelines. We test the proposed method on a wide range of TEM images, and the experimental results show that it can improve the detection accuracy significantly compared to methods that only use one kind of image information.

In Chapter 3, we present the retrospective analysis for the *in situ* TEM videos. After the particles are detected from the video, we estimate the time-varying PDF of nanoparticle size from all the video frames. The PDF is represented by a penalized B-spline model and the coefficients of the basis functions are estimated by the alternating direction method of multipliers (ADMM) [24]. Then a robust change point detection method is proposed to select the significant change points and identify possible growth stage changes during the

process. The detection results are consistent with the discoveries made in [1]. A sensitivity analysis shows that the proposed method is robust to the change of the tuning parameters. We build a hybrid model to describe the multi-stage growth process by combining the physics model of each stage. The hybrid model shows better accuracy than any single-stage growth models.

In Chapter 4, we introduce the work of the prospective analysis. With the B-spline representation, a state-space model is built to describe the change of the PDF of the particle sizes. Then a fast, closed-form expression is developed to update the particle size distribution online. While the use of a space state model ensures the temporal continuity of the estimation, we need another constraint to guarantee the curve's smoothness. For this purpose, we reformulate our state-space model and allow a penalty on the second-order difference of the state vector to be naturally incorporated in the new formulation. A Bayesian method is conducted to estimate the system parameters from the first several seconds of the video. We apply the proposed method to three different *in situ* TEM videos and obtain insightful tracking results of the nanoparticle growth status. In out-of-sample testing, we demonstrate that our estimation achieves better performance compared to those without the constraints of the curve smoothness or temporal continuity.

Finally, we summarize our major contributions in Chapter 5 and also discuss future extensions of the existing work in both application and methodology areas.

2. ROBUST NANOPARTICLES DETECTION FROM NOISY BACKGROUND BY FUSING COMPLEMENTARY IMAGE INFORMATION[1]

Chapter 2 studies the problem of detecting the presence of nanoparticles in noisy TEM images and then fitting each nanoparticle with an elliptic shape model. In order to achieve robustness while handling low contrast and high noise in the TEM images, we propose an approach to fuse two kinds of complementary image information, namely the pixel intensity and the gradient (the first derivative in intensity). Our approach entails two main steps: (a) the first step is to, after necessary pre-processing, employ both intensity-based information and gradient-based information to process the same TEM image and produce two independent sets of results; (b) the subsequent step is to formulate a binary integer programming (BIP) problem for conflict resolution among the two sets of results. Solving the BIP problem determines the final nanoparticle identification. We apply our method to a set of TEM images taken under different microscopic resolutions and noise levels. The empirical results show the merit of the proposed method: it can process a TEM image of 1024×1024 pixels in a few minutes, and the processed outcomes appear rather robust.

2.1 Introduction

As more and more nanoparticle-embedded materials are moved from labs to commercial use, we witness an increasing need for automated nanoparticle detection and characterization based on the electron microscopic images of nanoparticles [1, 2, 25]. The images include those from both TEM and SEM. Once the images are processed, material scientists would like to characterize the morphology of nanoparticles, or to quantify the

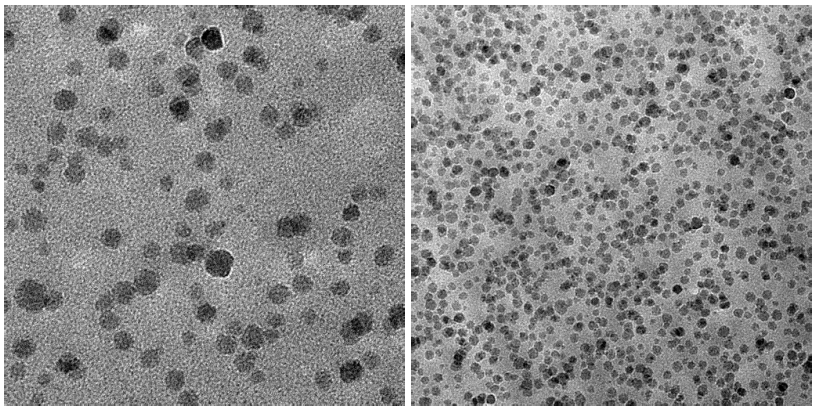
[1]Reprinted with permission from Y. Qian, J. Z. Huang, X. Li, and Y. Ding, "Robust nanoparticles detection from noisy background by fusing complementary image information," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5713–5726, 2016. https://doi.org/10.1109/TIP.2016.2614127,

dispersion of nanoparticles in the host material, as both traits are believed to have profound impact on the final material properties [26, 27]. To achieve these goals, the first job is to locate individual nanoparticles as accurately as possible, and then to characterize the shape and size of the nanoparticles. As such, automated detection and characterization of nanoparticles play important roles on nanomaterial exploration and production.

Park et al. [9] summarized the challenges associated with detection and characterization of nanoparticles from TEM images. The challenges lie in the facts that the nanoparticles are numerous and overlapped, and the variety of their shapes and sizes is also large. Park et al. [9] reviewed a number of image processing methods, including watershed transforms with different stopping criteria [28, 29], sliding band filter [30], graph cut [31], active contour [32], iterative voting [33], and a multiscale morphological method (a sophisticated variant of watershed) [34]. They argued that these methods cannot be directly applied to the TEM images due to the technical challenges mentioned above. Park and his colleagues [9, 7] proposed image processing and shape analysis approaches, tailored to nanoparticle image processing. There are also some recent developments on detecting and measuring nanoparticles in TEM images. Yang and Ahuja [35] proposed a segmentation method to isolate the granular objects using a local density clustering and gradient barrier watershed. De Temmerman et al. [36] designed a semi-automatic approach to measure the size of the primary particles in the TEM images of powdered nanomaterials, also relying on the watershed transfer for segmentation. Muneesawang and Sirisathitkul [37] proposed a multi-level segmentation method for identifying nanoparticles. They applied a k-means method to segment the TEM image into several layers and then produced multiple binary images associated with different thresholds. After that, they separated particles by applying the watershed method to each binary image and merged the results to remove over-segmentations. Overall, these developments advance the state of the art in handling TEM images for material characterization and exploration. However, when we try to ap-

ply these methods to a set of TEM images at hand, the resulting performances are not satisfactory. In particular, the quality of detection and characterization is not robust under different resolutions and noise patterns. To understand the reason, we should discuss the difference between the TEM images we have and the images processed in those works.

Our TEM images are taken from an important kind of nanocomposite, which has bisphenol-F epoxy resin as the host material, blended with silicon dioxide nanoparticles through a sol-gel process [38]. With epoxy resin as the polymer matrix and nanosilica as the nanofiller, it has enhanced mechanical properties such as modulus, hardness and fracture toughness while maintaining the optical properties (e.g., transparency), and is widely used in both academic research and in industrial applications [39]. To attain a TEM image for such material, one typically takes a thin slice of sample, which has the thickness of 50 to 100 nanometers (nm) and is transparent to naked eyes. The slice is thin enough for electrons to pass through, producing an image. Two examples of the TEM images, under different instrumental resolutions, are shown in Figure 2.1, labeled as “F3-2_7" and “F10_8", respectively. In the images, the darker dots represent the nanoparticles, whereas the gray background represents the host material.



(a) F3-2_7 (b) F10_8

Figure 2.1: Two examples of the TEM images of silica nanoparticles.

Compared with the nanoparticle images processed in [9, 7, 35, 36, 37], the TEM images at our disposal have much lower contrast and higher level of noises. This is due to the fact that our nanoparticles are silica particles, whose ability of shielding off or bouncing back electrons from passing through is weaker than metallic particles, and the silica particles are blended in a solid host material, whose density is not much less than the particles themselves. Meanwhile, the nonuniformity of the resin makes the background uneven (see Fig. 2.12(d)). By comparison, the nanoparticles used in most of the above-referenced works are metal ones, e.g., Au in [9, 7], Ti in [36] and FePt in [37]. Those metal particles, considering its large mass, are particularly potent in bouncing back electrons, producing a sharp contrast between the particles and the background. The noisy nature of our TEM images makes the detection and characterization task more challenging. In this chapter, our focus is to develop a new method for image segmentation targeting on the nanoparticle detection problem in noisy and low contrast TEM images.

There are two kinds of information commonly used for image segmentation: the intensity information and the gradient information [40, 41]. The intensity-based approach is to classify the pixels with similar intensity to the same category (be it an object or the background). The gradient-based information is calculated as the first-order derivative of the pixel intensity, signaling the magnitude of change along the way. A large gradient implies an edge (or a boundary, or a contour) separating an object from the background. To our best knowledge, many existing nanoparticle detection methods, for instance, [7, 42, 36, 37] among others, make primary use of one kind of image information (the use of intensity is more popular), causing them to only work well in certain circumstances. A natural remedy for that problem is an effective use of both kinds of image information.

The desire of combining the two kinds of image information has been raised. One strategy of combining information is to use different kinds of information sequentially, i.e. amend or enhance the segmentation results coming from the one kind of information by the

other. For example, the method in [43] first over-segments an object based on the intensity and then subsequently removes false boundaries by using gradient information. Another example is the boundary refinement method [44], in which the initial boundary of an object comes from the intensity information, and the boundary is then adjusted by taking into account the gradient of pixel intensity. In the recent development, the sequential strategy is also tailored to handle specifically nanoparticle images. For instance, the approach in [9] is to first segment the foreground based on pixel intensity and find the location of centers via a modified watershed transform [28]. Then, a center is matched with the edge/boundary of the same particle, produced by Canny's edge detector [45]. At last, the approach in [9] combines the two image features (center and boundary) to locate each particle. This type of information-combining approaches work well when the boundaries detected based on gradient are similar to that of the intensity-based results, to make sure that combining the two kinds of information through a compromise could produce a better result. However, those approaches are not applicable to the noisy TEM images, since the segmentation outcomes produced by using each kind of image information alone can be drastically different (refer to Figure 2.8 (a) and (b)), leaving little common ground for a compromise.

Another strategy of information combining is to design an energy functional, say the Mumford-Shah functional [46], integrating both kinds of information. Then the boundary of the foreground is evolving to maximize the energy functional until the local optima is found; doing so is supposed to produce the optimal separation [47]. To use the intensity information, Chan and Vese [32] assume that inside (or outside) the boundary, the variance of intensities of image should be small and Li et al. [48] assume that the intensities of pixels should change gradually, whereas to use the gradient information, Caselles et al. [49] assume that the gradient of the image along the boundary should be strong. Many recent works [50, 51, 52] consolidate these assumptions and design their versions, which,

to certain extent, make use of both intensity and gradient information. However, applying this strategy alone cannot handle the segmentation problem of overlapped nanoparticles, because nearly all such methods, including [32, 52], can only segment the foreground from the background, leaving the overlapped objects intact within the foreground. Methods considering multiple objects detection, such as [48], require the objects at the foreground to have unconnected boundaries, namely that the multiple objects cannot overlap.

Recognizing the shortcomings (and strengths) of those strategies, we propose a new framework to fuse the two kinds of image information via a parallel approach. Our approach starts off with focusing the two kinds of image information separately on the same TEM image. In other words, a TEM image is handled by two pipelines of processing in parallel. One pipeline is using primarily the intensity information, segmenting the foreground by a k-means clustering [53] and then separating the particles according to the shape of the foreground found by a watershed transform [28], whereas the other pipeline is using primarily the gradient information, going through an active contour [32] procedure to find the foreground, followed by an iterative voting method [33] that finds the center of each particle. Intensity information is also used in the second pipeline but the main driving force therein is the gradient information, differentiating it from the first pipeline of processing.

The two pipelines of processing produce two sets of outcomes for the same image, and as expected, some of the particle detection outcomes agree with each other (which means two detections by different methods are almost the same), while many others do not. When the detection outcomes agree, it reinforces the belief that they both indicate a good detection, and when the detection outcomes differ, we then need to resolve the conflict and choose one of the outcomes. Based on a fitness criterion to be introduced later, we select the particle detection with the highest fitness score and discard those conflicting with it. A binary integer programming (BIP) is formulated and solved to obtain the optimal solution.

In order to handle TEM images containing numerous particles, we also accelerate the optimizing procedure by using a subgraph decomposition technique. Our framework is illustrated in Figure 2.2.

We want to note a similarity between our fitness score approach and that in [54], which is in the context of tree detection. The approach in [54] is based on random point process and can be seen as a soft version of the optimization problem formulated in our method, where overlapping is penalized but not forbidden. The random point processes are solved through Markov chain Monte Carlo, which is rather complex to optimize than the BIP formulation used in our approach.

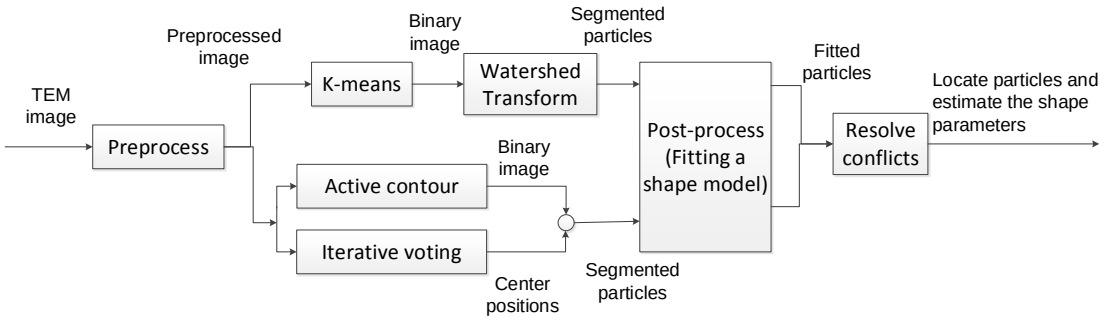


Figure 2.2: The two pipelines of processing to make use of the complementary image information.

The remaining parts of the chapter are organized as follows. In Section 2.2, we describe the basic thoughts behind the choice of the components in each processing pipeline. In Section 2.3, we present the formulation and solution that resolves the conflicts between the two sets of processing outcomes. In Section 2.4, we apply our method to a set of TEM images, obtained under different instrumental resolutions and noise conditions, and assess the method's effectiveness and efficiency. Finally, we summarize our work in Section 2.5.

2.2 Basic Components in Processing Nanoparticle Images

The processing of nanoparticle images, illustrated in Figure 2.2, consists of three main steps: the preprocessing, the two pipelines of processing, and the postprocessing. This section intends to provide an overview of the basic components in the proposed framework.

The preprocessing is to enhance the image features from the noisy raw images, while the postprocessing is to fit a parametric shape model, once a nanoparticle is isolated. The two pipelines of image processing in between intend to locate the nanoparticles and isolate each of them as accurately as possible.

The two pipelines of processing are carried out on the same image in parallel. Specifically, one pipeline of processing uses primarily the intensity information, whereas the other uses primarily the gradient information. Each pipeline further involves two methods for separating and identifying the nanoparticles.

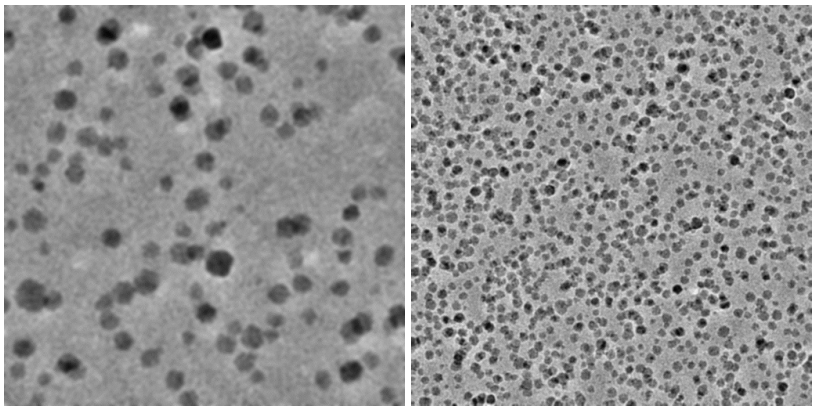
In this framework, many existing methods are used. In order to produce better results, however, certain methods, especially those used in the two pipelines of processing, are tailored towards the uniqueness of TEM images.

2.2.1 Preprocessing

The preprocessing intends to strengthen the contrast of the nanoimages and remove the unevenness in background. The background unevenness is a result of having non-uniform thickness in the slice of resin samples. Consequently, the resulting images are usually lighter on one corner/side and darker on the opposite corner/side; please see Figure 2.1 for an example. Two operations are used in the preprocessing: Butterworth high-pass filtering (4th order) [55] and Gaussian filtering [56].

Butterworth high-pass filtering removes the low frequency components of the image (related to the unevenness in background). A low frequency cutoff is used to identify background patterns of large size, supposedly far greater in size than a typical particle. We

set this value as $1,024$ divided by four folds of the average particle diameter, where the factor of four is chosen empirically. Meanwhile, using a Gaussian filter intends to remove the high frequency components, weeding out the small objects that cannot possibly be a particle. The parameter in the Gaussian filter is set to be one-tenth of the nanoparticle's average diameter. By linking the filtering strength to the particle's average diameter, the strength of Gaussian filter's smoothing strength can be adaptively adjusted. We show the results of the preprocessing in Figure 2.3.



(a) F3-2_7 (b) F10_8

Figure 2.3: The preprocessing results of the two image examples in Fig. 2.1.

2.2.2 Intensity-Based Processing

In the first pipeline of processing, pixel intensity is used. This line of processing entails two steps: the first step is a k-means method [57] to separate the foreground from the background, producing the nanoparticle agglomerates, namely nanoparticle clusters. The second step is to use the watershed transform on the segmented foreground that further breaks the overlapped particles in the nanoparticle agglomerates into individual particles.

Each pixel in the first step is classified based on not only its intensity but also its

coordinates. In [58], the image data is converted into a 5-dimensional vector $[x, y, l, u, v]$ for each pixel, where the x, y are the coordinates and l, u, v are the color values in LUV color space. For our gray-level TEM image, the input vector is set as $[w_s x, w_s y, R]$, where R is the intensity and w_s is a weighting coefficient to balance the effects between the coordinate value and the image intensity value. In this work, we set $w_s = 0.2$, as it is close to the ratio of the largest grayness value over the size of the TEM image, so we will have the similar ranges of the three coordinates. Then, we seek to find $k = 2$ clusters among the image pixels, corresponding to the foreground and background, respectively.

The second step is a watershed transform based on the shape of the foreground. A watershed transform goes through an erosion-dilation cycle, in which erosion produces the cores of neighboring objects (called markers) and dilation identifies the separating boundary lines between the objects. The specific variant of watershed transform we adopt is the Ultimate Erosion for Convex Sets (UECS) proposed by Park et al. [9], which tailors its erosion stopping criterion towards convex objects, as the physical-chemical forces behind nanoparticle formation do drive nanoparticles to have convex shapes.

In the implementation of the UECS, we found that the number of erosion steps can vary widely on different particles. One shortcoming of this variation in erosion steps is that the separating lines between particles tend to over-erode one of the particles. This phenomenon is illustrated in Figure 2.4(a); please note the over-erosion of boundary lines inside particle 2 and particle 3.

The remedy we devise to alleviate the over-erosion problem is to record the number of erosion steps, following a generic idea first introduced in [34]. Provided the number of erosion steps associated with each particle, the dilation process is then timed following the descending order of the number of erosion steps that had been performed on respective particles. For instance, suppose particle 1 was eroded 10 times to its final marker, while particle 2 was eroded 20 times. In dilation, we start with particle 2 and dilate its marker

10 times, and at which time, start the dilation of the marker of particle 1 in parallel, until the two dilated markers meet. It appears that this simple revision improves the accuracy of the boundary lines between particles appreciably; please see Figure 2.4(b).

Figure 2.4: The comparison of two watershed segmentations: (a) the result of the original UECS; (b) the result of the revised UECS with a timed erosion-dilation process.

2.2.3 Gradient-Based Processing

Gradient-based processing makes use of the gradient of an image to detect and separate the nanoparticles. As mentioned before, gradient-based processing also uses pixel intensity information; it is just that the gradient information plays a more deciding role here. This line of processing also entails two elements: an active contour method [52] that is based on the level set formulation and the iterative voting method [33].

The active contour method identifies the boundary (or edge) for the nanoparticles, without necessarily separating a particle agglomerate into individual particles. Then the iterative voting method locates the centers of individual particles from the preprocessed TEM images. Once the centers of individual particles as well as the boundaries of particle agglomerates are available, the connected particle contours can be separated and then assigned to individual particles by using an edge-to-marker association technique, similar to what was initially proposed in [9].

One difficulty of using the active contour method for the low-contrast, noisy TEM images is that the boundary of particles is blurred and the background is noisy, making the convergence of the recursive method sensitive to the choice of the initial contour (also known as a mask). We propose two remedies to ensure a robust convergence. Firstly we choose the active contour method, proposed by Tian et al. [52], that uses both intensity and gradient information, as it has a better convergence property than its counterparts that use only the intensity or the gradient information (e.g., [32, 49]).

Even with a capable method like [52], the choice of the initial contour still has a profound impact on the outcomes of contour detection for the silica nanoparticles. To find a proper initialization, we start the active contour method from either a large mask or a small mask of the foreground. Then the active contour algorithm can shrink the large mask or expand the small one to get the estimated contour. To see which mask leads to good detection outcome, we first apply Otsu's method [59] to get a binarization threshold R_t. And then we select an offset value R_s, so that we can choose masks of different sizes. Then the large mask, denoted by M_1, can be obtained by $M_1 = \{(x, y)|R(x, y) < R_t + R_s\}$, whereas the small mask, denoted by M_2, by $M_2 = \{(x, y)|R(x, y) < R_t - R_s\}$. Once M_1 and M_2 are used, the convergent outcomes are denoted by B_1 and B_2, respectively. We find that with Tian et al. [52]'s algorithm, B_2 (expanded from a small mask) is much better than B_1 (shrunk from a large mask); see an example in Figure 2.5. We believe that the noisy background of the TEM images makes the use of large masks ineffective (algorithm trapped in local optima). Therefore, we choose the small mask as the initial contour in the chosen active contour method.

Concerning the use of the iterative voting method, we also tailor the original method in [33] to our nano imaging problem. In the original approach, Parvin et al. [33] choose the pixels on the edge that are detected by Canny's edge detector [45] and use them to vote for locating the centers. The problem with this approach is that when some edges

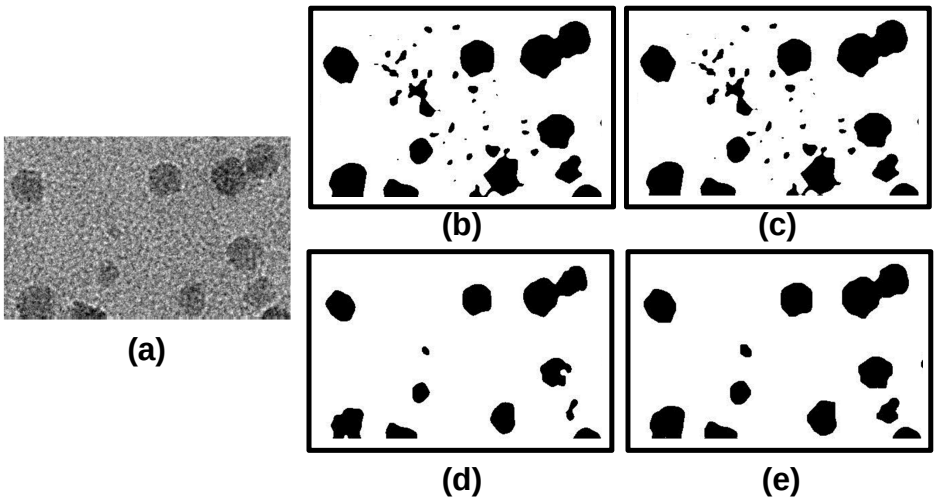
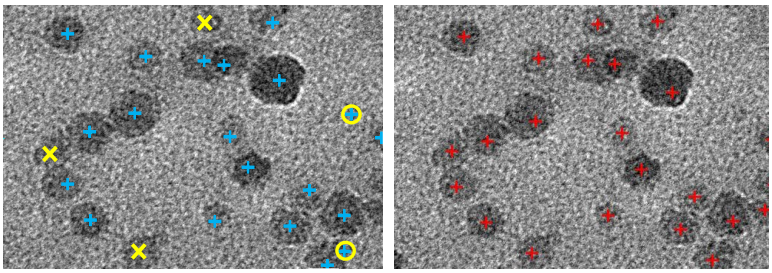


Figure 2.5: The process of active contour with different initializations: (a) the original image; (b) the large mask M_1; (c) the convergent result B_1 from the large mask M_1; (d) the small mask M_2; (e) the convergent result B_2 from the small mask M_2.

are hard to detect, such as in our noisy nanoparticle images, some nanoparticles will be missed. Figure 2.6(a) shows that using the original iterative voting method in a small region of about 20 particles produces three misses and two false detections; to produce Figure 2.6(a), we use the `ImageJ` plugin of the iterative voting method [33].

Our tailoring works as follows. Note that a large magnitude of gradient indicates that the corresponding pixel is more likely to be on the edge. We hence select all pixels of the preprocessed image whose gradient is larger than a threshold in magnitude, and deem them as our potential voter pixels. We then set the weight of each voter proportional to its magnitude. The threshold here is chosen as one-fifth of the maximal magnitude of gradient in the whole image. Because we skip the action of Canny's edge detection, we could not initialize the iterative voting procedure using the normal direction to the detected edges, as recommended in [33]. Instead, we let the voting direction initialized as opposite to the gradient direction at a voting pixel. Figure 2.6(b) shows the outcomes of our tailored

approach, which is carried out on the same image and produces a result without misses and false detections.



(a) ImageJ Plugin (b) Our Implementation

Figure 2.6: The results of iterative voting: (a) `ImageJ` plugin outcome, where the three misses are indicated by yellow X's and two false detections is marked by yellow circles; (b) our implementation outcome.

2.2.4 Postprocessing

In post-processing, we fit each identified nanoparticle with a parametric shape model. Unlike in [9] where a particle is modeled by a B-spline, our treatment here is much simpler – we use an elliptic shape model that can be parameterized using five parameters $[x_0, y_0, a_0, b_0, \theta_0]$ (Figure 2.7(a)), where x_0 and y_0 are the coordinates of the center, a_0 and b_0 are the lengths of the long and short axes, and θ_0 is the orientation of the particle. The reason that we choose a simple shape model is that the silica nanoparticles produced by the sol-gel process are mostly of round or ellipse shapes; by contrast, the nanoparticles processed in [9] have shapes of wider varieties. In the meanwhile, given the noise level in the nanoimages processed in this work, it becomes less robust to use complicated shape models with too much flexibility, as a flexible shape model may be too eager to adapt itself to background noises surrounding a particle.

When it comes to the fitting of an elliptical shape, we choose to use the second-moment fitting method [60], which finds an ellipse that has the same mass center and same second moments as those of a detected particle region. This treatment uses all the pixels inside the contour of a detected particle, rather than rely on the detected contour of a particle. The drawback of using the detected contour alone is its sensitivity to shape noises, because many detected contours can end up with an irregular shape; see the example in Figure 2.7(a) (the gray region). This second-moment method produces much more robust shape fitting outcomes, as evident by the comparison between Figure 2.7(b) and (c).

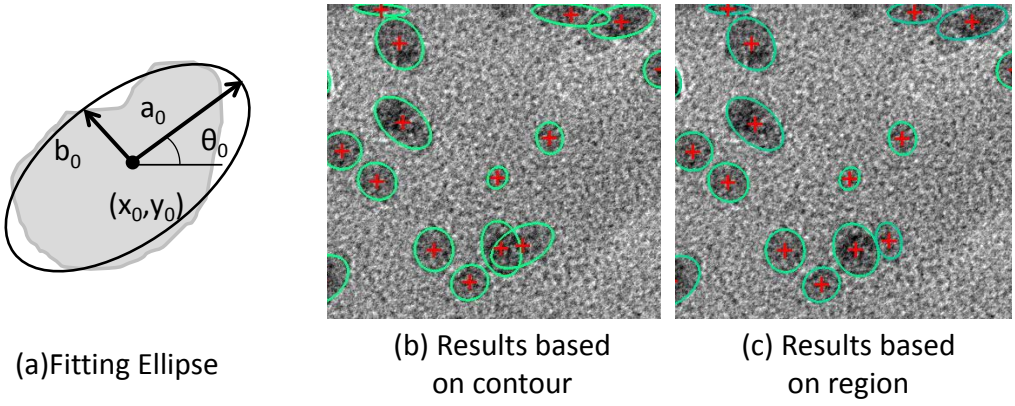


Figure 2.7: Post-precessing: (a) parametrization of an elliptical shape; (b) the fitting outcomes based on contour alone; (c) the fitting outcomes based on all the pixels in a detected particle region.

2.2.5 Pros and Cons of the Two Pipelines of Processing

In Figure 2.8, we highlight four examples to illustrate the pros and cons of the two pipelines of processing. In example $\#1$, the gradient-based processing produces a better boundary of the right-side particle than the intensity-based processing does. In example $\#2$, the iterative voting in the gradient-based pipeline successfully segments two over-

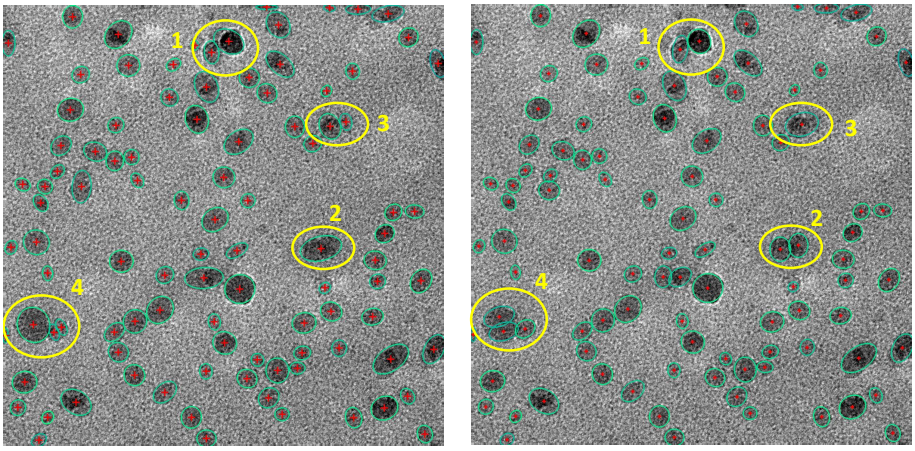
lapped particles based on the intensity change inside the foreground region, whereas the intensity-based processing fails to do so. In example $\#3$, the gradient-based processing fails to identify the right-side particle because of the blurred boundary, whereas the intensity-based process does detect. In example $\#4$, the gradient-based processing over-segments the left-side particle, whereas the intensity-based processing over-segments the right-side one.

Generally speaking, our observations suggest that when the gradient is clear and accurate, the gradient-based processing works better ($\#1$ and $\#2$); otherwise the intensity-based processing will be more robust ($\#3$). For some harder cases, such as $\#4$, each pipeline of processing does half right, so only combining the two sets of the results can further improve the accuracy of the final detection. While the general observations make intuitive sense, it is not always so easy to tell which pipeline of processing will do better under a specific circumstance. This implies that the criterion that gradient information is clear and accurate sometimes can be difficult to assess and quantify manually. What is needed is an automatic confict resolution procedure that can pick the better of the two detection outcomes.

2.3 Fusing the Complementary Information

The next step is to make use of the detection results from the two pipelines of image processing and produce an enhanced detection outcome. The problem is similar to multi-expert decision making [61], where the two pipelines of detection act as two experts and the sets of detected particles are their decisions. If both experts agree with each other on all decisions, then the problem is trivial, as one can choose either set of the outcomes. Otherwise we should devise a conflict resolving procedure to choose one of them or discard both.

Let us first introduce some notations. Let $I = \{I(i), i = 1, \cdots, N_I\}$ and $G =$



(a) Intensity-based Pipeline (b) Gradient-based Pipeline

Figure 2.8: The comparison of the results of the intensity-based and the gradient-based processing.

$\{G(j), j = 1, \cdots, N_G\}$ denote the detected particles, respectively, by the intensity-based and gradient-based pipeline, where N_I and N_G are the corresponding numbers of particles detected.

The five shape parameters of $I(i)$, defined in Section 2.2.4, are expressed as $[x_0(I(i)), y_0(I(i)), a_0(I(i)), b_0(I(i)), \theta_0(I(i))]$. The set of pixels within the fitted ellipses is labeled as $P_{I(i)}$, and its cardinality $|P_{I(i)}|$ represents the area of the corresponding region. The corresponding notations for $G(j)$ can be defined similarly.

We use the binary variables $b_{I(i)}$ and $b_{G(j)}$ to indicate the outcome of our resolution: if $I(i)$ (or $G(j)$) is chosen as the final detection outcome, then $b_{I(i)}$ (or $b_{G(j)}$) will be set as 1, otherwise it is set as 0. Aggregating all the decision variables associated with individual detections, the decision vector for the intensity-based approach is expressed as $\mathbf{b}_I = [b_{I(1)}, \cdots, b_{I(N_I)}]^T$, and that for the gradient-based approach is $\mathbf{b}_G = [b_{G(1)}, \cdots, b_{G(N_G)}]^T$. Our goal is to find an optimal solution of $\mathbf{b}_I$ and $\mathbf{b}_G$, which is to properly set elements of $\mathbf{b}_I$ and $\mathbf{b}_G$ to 1 or 0, according to an optimality criterion introduced below.

2.3.1 Basic Formulation for Conflict Resolution

One crucial step in making good use of the two types of detection outcomes is to understand the three possible relationships between $I(i)$ and $G(j)$. The relationship is illustrated in Figure 2.9. When two detection outcomes have only a slight overlap or no overlap at all, as shown in Column (a) of Figure 2.9, it is unlikely that they are related to the common particle in the image. When the two outcomes virtually coincide with each other, manifesting in a heavy overlap between the detection regions, they point to the same underlying particle and are then referred to as a consensus detection. When the two outcomes occupy the same region in the image, but the detected particles have serious disagreement, either in number (one approach detects one particle, while the other detects two, for instance) or in key shape parameters (including the center location), these outcomes are referred to as the conflicting detections. The consensus detections and the conflicting detections are illustrated in Columns (b) and (c) of Figure 2.9, respectively. The unrelated and consensus detections are relatively straightforward to deal with. It is the conflicting detections that need further processing to decide which one to be the final detection outcome.

As such, there are two primary questions to be addressed:

1. How to determine which category of relation (unrelated, consensus, and conflicting) $I(i)$-versus-$G(j)$ belongs to?

2. Once this relation is determined as a conflicting detection, what criterion to use to make the final selection?

The answer to the first question apparently depends on the degree of overlap between two detections; the above description of the three relationships provides the intuition behind it. The specific formula will be presented later in Section 2.3.2.

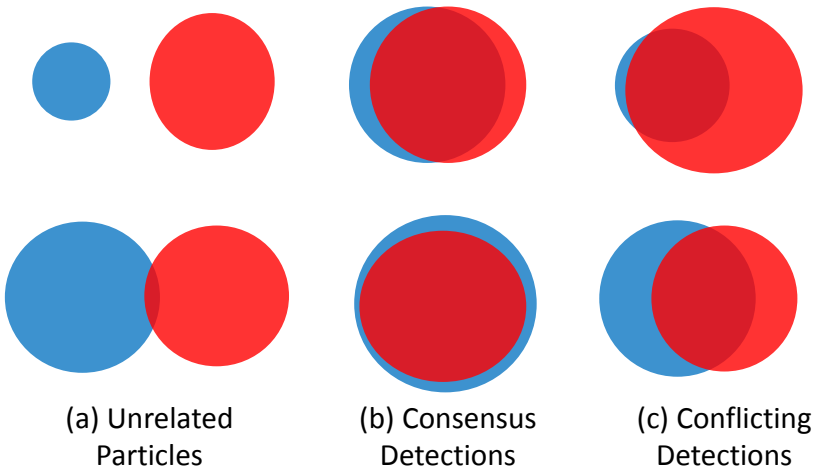


Figure 2.9: Three possible relationships between $I(i)$ (blue) and $G(j)$ (red): (a) two detection results are not related to the same particle; (b) two results coincide with each other; (c) two results are in conflict.

To address the second question, we assign each particle detection with a score, assessing its fitness to the original image. Intuitively speaking, the higher the score, the better a detection fits the original image. We denote the fitness score vector of a detection as $\mathbf{s}_I = [s_{I(1)}, \cdots, s_{I(N_I)}]^T$ for the intensity-based approach and $\mathbf{s}_G = [s_{G(1)}, \cdots, s_{G(N_G)}]^T$ for the gradient-based approach. The specific definition of the fitness score is provided in Section 2.3.3.

When the two pipelines of processing reach a consensus, it enhances the credibility of the detection and makes such detection more reliable and trustworthy. It is safe to take the consensus outcomes and add them into the final detection results without further processing. We compute the shape parameters of the final particle by averaging the corresponding parameters of the two detections. Then we remove these particles from the sets of I and G, so that only the conflicting detections are left to be resolved. Denote the sets of the remaining particles as $\tilde{I} = \{\tilde{I}(1), \cdots, \tilde{I}(N_{\tilde{I}})\}$ and $\tilde{G} = \{\tilde{G}(1), \cdots, \tilde{G}(N_{\tilde{G}})\}$, where $N_{\tilde{I}}$

and $N_{\tilde{G}}$ are the numbers of particles in the two revised sets, respectively. In the subsequent conflict resolving procedure, we only need to solve for $\mathbf{b}_{\tilde{I}}$ and $\mathbf{b}_{\tilde{G}}$, which are a subset of $\mathbf{b}_I$ and $\mathbf{b}_G$, respectively, and have usually fewer than half of the original elements.

For the remaining conflicting detections, we use a conflict matrix $\mathbf{M} = (M_{ij})$ to connect them. $\mathbf{M}$ is an $N_{\tilde{I}} \times N_{\tilde{G}}$ binary matrix, with each row representing one particle in $\tilde{I}$ and each column representing one particle in $\tilde{G}$. If $\tilde{I}(i)$ and $\tilde{G}(j)$ are conflicting, $M_{ij} = 1$; if they are unrelated, $M_{ij} = 0$. Figure 2.10 shows a simple example of conflicting detections and the corresponding conflict matrix. In Figure 2.10, we observe that $\tilde{I}(1)$ is conflicting with $\tilde{G}(1)$, while $\tilde{I}(2)$ is conflicting with both $\tilde{G}(2)$ and $\tilde{G}(3)$; this is reflected in the 2×3 conflict matrix to the right.

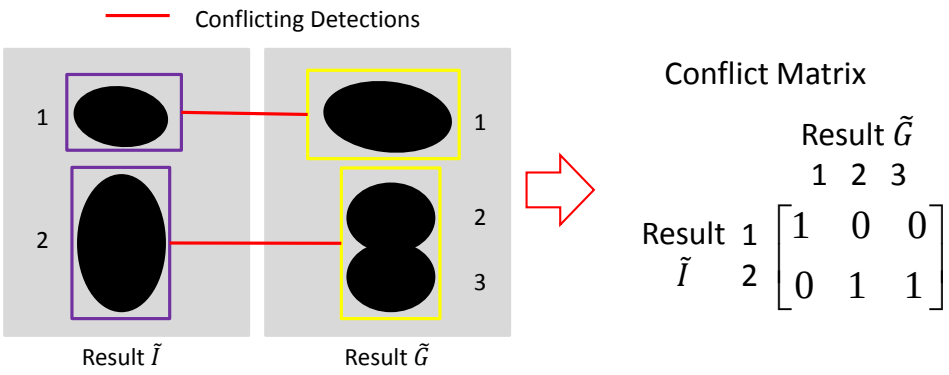


Figure 2.10: An example of conflicting detections (left) and the corresponding conflict matrix (right).

With a fitness score chosen, we present the following constrained binary integer programming (BIP) problem for selecting the final detection out of a conflict:

$$\begin{aligned} &\max_{\mathbf{b}_{\tilde{I}}, \mathbf{b}_{\tilde{G}}} \quad \mathbf{s}_{\tilde{I}}^T \mathbf{b}_{\tilde{I}} + \mathbf{s}_{\tilde{G}}^T \mathbf{b}_{\tilde{G}}, \\ &\text{subject to} \quad \mathbf{b}_{\tilde{I}}^T \mathbf{M} \mathbf{b}_{\tilde{G}} = 0. \end{aligned} \tag{2.1}$$

The objective function is the summation of the fitness scores of all detections, and we aim to obtain the highest total fitness score for an image. The constraint function is to ensure that only one of the conflicting detections will be chosen. To see this, rewrite the constraint as

$$\sum_{i=1}^{N_{\tilde{I}}} \sum_{j=1}^{N_{\tilde{G}}} b_{\tilde{I}(i)} M_{ij} b_{\tilde{G}(j)} = 0, \tag{2.2}$$

meaning that if $\tilde{I}(i)$ and $\tilde{G}(j)$ are a pair of conflicting detections, namely $M_{ij} = 1$, then $b_{\tilde{I}(i)}$ and $b_{\tilde{G}(j)}$ cannot be 1 simultaneously. We solve this BIP problem by using the `MATLAB` solver '`bintprog`'. To use it, we multiply a negative sign to the objective function to change the problem to a minimization problem.

We want to note that the authors in [54] also used model fitness in an application of tree detection. More specifically, they first use an unknown number of ellipses to model the trees on a plantation and then calculate the prior energy and the likelihood according to prior knowledge and the observed images. At last, they minimized the Bayesian energy using Markov chain Monte Carlo to find the ellipses that fit the tree crowns the best. There are a couple of differences between their method and our conflict resolution approach. Firstly, the approach in [54] is based on random point process. They wanted to minimize the overlapping of different ellipses, which penalizes the overlapping but does not forbid it, whereas in our approach, we have to choose one of the outcomes. Secondly, their Bayesian based solution procedure is more complicated than the BIP formulation we use.

2.3.2 Consensus and Conflicting Detections

From Figure 2.9, we can see that the degree of overlap between the two detection outcomes can be used to decide which category a pair of detections belongs to. When the Euclidean distance between the centers of the two detections is larger than $[a_0(I(i)) + a_0(G(j))]/2$, it means that there is no overlap between the two detected particles. The pair is then unrelated. When the distance is smaller than $[a_0(I(i)) + a_0(G(j))]/2$, we need to

quantify the degree of overlap. The area of overlap is $|P_{I(i)} \cap P_{G(j)}|$. We calculate the maximum overlapping ratio $r_{\max}$ and minimum overlapping ratio $r_{\min}$ as follows:

$$\begin{aligned} r_{\max}(I(i), G(j)) &= \max\{\tfrac{|P_{I(i)} \cap P_{G(j)}|}{|P_{I(i)}|}, \tfrac{|P_{I(i)} \cap P_{G(j)}|}{|P_{G(j)}|}\}, \\ r_{\min}(I(i), G(j)) &= \min\{\tfrac{|P_{I(i)} \cap P_{G(j)}|}{|P_{I(i)}|}, \tfrac{|P_{I(i)} \cap P_{G(j)}|}{|P_{G(j)}|}\}. \end{aligned} \tag{2.3}$$

We then set two thresholds, an upper ratio r_U and a lower ratio r_L, such that if $r_{\max}(I(i), G(j)) < r_L$, we deem the overlapping region small enough to declare $I(i)$ and $G(j)$ unrelated; if $r_{\min}(I(i), G(j)) < r_U$ and $r_{\max}(I(i), G(j)) > r_L$, we believe that the two detection outcomes are related but different, namely that they form a pair of conflicts; if $r_{\min}(I(i), G(j)) > r_U$, we consider this as a consensus detection.

2.3.3 Fitness Score of Detections

Essentially, calculating the fitness score for each particle is equivalent to evaluating the quality of the image segmentation, in which a regional part of TEM images is separated into the particle and its surrounding area. Zhang et al. [62] surveyed different evaluation methods for image segmentation quality when the ground truth is unknown. They pointed out a simple principle that is still widely used: the inter-region disparity should be large and intra-region variability should be small. For instance, Fisker et al. [63] maximize the difference in the average intensities between the foreground and its surrounding background for detecting a particle. To measure the inter-region disparity and the intra-region similarity, we need to define a neighboring region Q for particles in $\tilde{I}$ and $\tilde{G}$. Consider a particle $\tilde{I}(i)$ (the same can be done to $\tilde{G}(j)$). Its foreground information is in $P_{\tilde{I}(i)}$ and the surrounding background information is in $Q_{\tilde{I}(i)}$ (shown in Figure 2.11). In identifying $Q_{\tilde{I}(i)}$, we double the size of $P_{\tilde{I}(i)}$, namely $|Q_{\tilde{I}(i)} \cup P_{\tilde{I}(i)}| = 2|P_{\tilde{I}(i)}|$, so that $|Q_{\tilde{I}(i)}| = |P_{\tilde{I}(i)}|$.

Our measure of the inter-region disparity and the intra-region similarity is based on the sum of squares of pixel intensities. The sum of squares are proportional to the variance

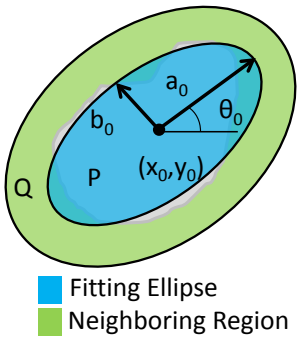


Figure 2.11: The foreground region P (blue) and its neighboring region Q (green) for a detected particle.

of the intensities within a region, so a large value indicates disparity while a small value indicates similarity. For a good segmentation, the sum of squares of the whole region should be much larger than that of separated background or foreground. For an arbitrary region A in the image, its sum of squares of the intensity, denoted by $SS(A)$, is calculated by:

$$SS(A) = \sum_{(x,y)\in A} [R(x,y) - \bar{R}(A)]^2, \tag{2.4}$$

where $\bar{R}(A)$ is the average intensity of all pixels inside A. We then define the fitness score of $\tilde{I}(i)$ as:

$$s_{\tilde{I}(i)} = SS(P_{\tilde{I}(i)} \cup Q_{\tilde{I}(i)}) - [SS(P_{\tilde{I}(i)}) + SS(Q_{\tilde{I}(i)})] - \lambda |P_{\tilde{I}(i)} \cup Q_{\tilde{I}(i)}|, \tag{2.5}$$

where the first term $SS(P_{\tilde{I}(i)} \cup Q_{\tilde{I}(i)})$ measures the inter-region disparity, and the second term $[SS(P_{\tilde{I}(i)}) + SS(Q_{\tilde{I}(i)})]$ measures the intra-region similarity. The greater their difference, the stronger indication it is to think that $\tilde{I}(i)$ is part of the particle's foreground. The third term is a noise filter. Its inclusion forces the difference between the inter-region disparity and the intra-region similarity to be great enough so as to qualify $\tilde{I}(i)$ as a genuine

particle, helping reduce the false detections in a noisy image. If $\tilde{I}(i)$ is a single unrelated particles, which means it has no conflicting detection in another set of results, it will be selected if and only if $s_{\tilde{I}(i)}$ is larger then 0.

In Equation (2.5), the first term is the total sum of squares of the whole region and the second term is the within-group sum of squares. According to the property of variance [64], their difference equals the between-group sum of squares, i.e.,

$$|P_{\tilde{I}(i)}|[\bar{R}(P_{\tilde{I}(i)}) - \bar{R}(P_{\tilde{I}(i)} \cup Q_{\tilde{I}(i)})]^2 + |Q_{\tilde{I}(i)}|[\bar{R}(Q_{\tilde{I}(i)}) - \bar{R}(P_{\tilde{I}(i)} \cup Q_{\tilde{I}(i)})]^2, \tag{2.6}$$

where $\bar{R}(P_{\tilde{I}(i)})$, $\bar{R}(Q_{\tilde{I}(i)})$ and $\bar{R}(P_{\tilde{I}(i)} \cup Q_{\tilde{I}(i)})$ are the average intensities of the foreground, its neighboring region, and the combined whole area, respectively. By the choice of neighboring region made above, namely $|Q_{\tilde{I}(i)}| = |P_{\tilde{I}(i)}|$ (they may not be exactly the same but the difference is negligible), it means:

$$\bar{R}(P_{\tilde{I}(i)} \cup Q_{\tilde{I}(i)}) = (\bar{R}(P_{\tilde{I}(i)}) + \bar{R}(Q_{\tilde{I}(i)}))/2. \tag{2.7}$$

Plugging in Equations (2.6) and (2.7) into Equation (2.5), we have

$$s_{\tilde{I}(i)} = |P_{\tilde{I}(i)} \cup Q_{\tilde{I}(i)}| \left\{ \left(\frac{\bar{R}(P_{\tilde{I}(i)}) - \bar{R}(Q_{\tilde{I}(i)})}{2} \right)^2 - \lambda \right\}. \tag{2.8}$$

It is now clear how the third term in Equation (2.5) works – if the intensity difference between the foreground and background is smaller than the threshold $2\sqrt{\lambda}$, then, the fitness score $s_{\tilde{I}(i)}$ turns negative, and consequently, $\tilde{I}(i)$ will not be chosen as a particle.

2.3.4 Decomposition and Linearization

To solve the optimization problem (2.1) efficiently, we need to address two more problems: (a) There are hundreds to thousands of particles in $\tilde{I}$ and $\tilde{G}$ in a TEM image. Solving

the optimization in its current form is time consuming. (b) The constraint in (2.1) is not linear, which prevents a straightforward application of some existing efficient methods. It is necessary to decompose the original problem into smaller-sized subproblems, and to linearize the constraint.

The way to decompose the original optimization problem is to decompose the conflict matrix $\mathbf{M}$. If $\mathbf{M}$ can be expressed in a block form with zero off-diagonal submatrices, then, each block submatrix can be used to form a separate BIP problem and be solved in parallel. A simple example is a two-block $\mathbf{M}$, such as

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 \end{bmatrix}, \tag{2.9}$$

then Equation (2.1) can be decomposed into two BIP problems:

$$\begin{array}{ll} \max_{\mathbf{b}_{\tilde{I}_1}, \mathbf{b}_{\tilde{G}_1}} \mathbf{s}_{\tilde{I}_1}^T \mathbf{b}_{\tilde{I}_1} + \mathbf{s}_{\tilde{G}_1}^T \mathbf{b}_{\tilde{G}_1} & \max_{\mathbf{b}_{\tilde{I}_2}, \mathbf{b}_{\tilde{G}_2}} \mathbf{s}_{\tilde{I}_2}^T \mathbf{b}_{\tilde{I}_2} + \mathbf{s}_{\tilde{G}_2}^T \mathbf{b}_{\tilde{G}_2} \\ \text{subject to } \mathbf{b}_{\tilde{I}_1}^T \mathbf{M}_1 \mathbf{b}_{\tilde{G}_1} = 0 & \text{subject to } \mathbf{b}_{\tilde{I}_2}^T \mathbf{M}_2 \mathbf{b}_{\tilde{G}_2} = 0, \end{array} \tag{2.10}$$

where $\mathbf{s}_{\tilde{I}} = [\mathbf{s}_{\tilde{I}_1}; \mathbf{s}_{\tilde{I}_2}]$ and $\mathbf{s}_{\tilde{G}} = [\mathbf{s}_{\tilde{G}_1}; \mathbf{s}_{\tilde{G}_2}]$. After solving those two subproblems, the minimizer of the original problem can be easily obtained by combining their individual solutions, namely $\mathbf{b}_{\tilde{I}} = [\mathbf{b}_{\tilde{I}_1}; \mathbf{b}_{\tilde{I}_2}]$ and $\mathbf{b}_{\tilde{G}} = [\mathbf{b}_{\tilde{G}_1}; \mathbf{b}_{\tilde{G}_2}]$.

The decomposition of the BIP can also be seen as a problem to find the connected independent subgraph. We regard the $N_{\tilde{I}} + N_{\tilde{G}}$ particles in $\tilde{I}$ and $\tilde{G}$ as nodes to build an undirected graph $\mathbf{G}$. Then, we connect two nodes if they form a fair of conflicting detection and obtain the corresponding adjacent matrix $\mathbf{W}$. If we can find an independent connected subgraph containing, for example, $\tilde{I}(1)$, $\tilde{I}(2)$ and $\tilde{G}(1)$, $\tilde{G}(2)$, $\tilde{G}(3)$, that means there is no conflicting relationship between them and any other particles. So we can form a subproblem only concerning those five particles, and the solution of that subproblem is the

same as the corresponding part of the whole problem. To find all connected independent subgraphs in $\mathbf{G}$, we adopt the spectrum analysis method in [65].

The theory in [65] says that the number of independent connected subgraphs of $\mathbf{G}$ equals to the multiplicity of 0 eigenvalue of its normalized graph Laplacian matrix:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}, \tag{2.11}$$

where $\mathbf{W}$ is the adjacent matrix of the graph $\mathbf{G}$, $\mathbf{I}$ is the identical matrix which has the same size of $\mathbf{W}$, and $\mathbf{D}$ is the diagonal matrix of the row (or column) sum of $\mathbf{W}$. [65] provides a detailed procedure. Following their procedure, first check if the graph $\mathbf{G}$ is decomposable (i.e., check the multiplicity of 0 eigenvalue of $\mathbf{L}$), and if this multiplicity is $K > 1$, then $\mathbf{G}$ can be decomposed to a set of K independent connected subgraphs. Then we can break $\mathbf{M}$ into K block submatrices $\{\mathbf{M}_k\}_{i=1}^K$, and the fitness score vectors $\mathbf{s}_{\tilde{I}}$ and $\mathbf{s}_{\tilde{G}}$ into $\{\mathbf{s}_{\tilde{I}_k}\}_{i=1}^K$ and $\{\mathbf{s}_{\tilde{G}_k}\}_{i=1}^K$, respectively. As such, the original BIP can be decomposed to K smaller subproblems that can be solved in parallel. The kth subproblem is:

$$\begin{aligned} &\max_{\mathbf{b}_{\tilde{I}_k}, \mathbf{b}_{\tilde{G}_k}} \quad \mathbf{s}_{\tilde{I}_k}^T \mathbf{b}_{\tilde{I}_k} + \mathbf{s}_{\tilde{G}_k}^T \mathbf{b}_{\tilde{G}_k}, \\ &\text{subject to} \qquad \mathbf{b}_{\tilde{I}_k}^T \mathbf{M}_k \mathbf{b}_{\tilde{G}_k} = 0. \end{aligned} \tag{2.12}$$

Next, we show that the constraint in Equation (2.1) can be linearized. Because $\mathbf{b}_{\tilde{I}}$, $\mathbf{b}_{\tilde{G}}$ and $\mathbf{M}$ are binary vectors/matrix, the original constraint can be replaced by the following inequality:

$$\mathbf{M}^T \mathbf{b}_{\tilde{I}} + N_{\tilde{I}} \mathbf{b}_{\tilde{G}} \leq N_{\tilde{I}} \mathbf{1}_{N_{\tilde{G}}}, \tag{2.13}$$

where $\mathbf{1}_{N_{\tilde{G}}}$ represents an $N_{\tilde{G}} \times 1$ vector whose elements are all 1's.

We can show that the original constraint and Equation (2.13) are equivalent. For the constraint in (2.1), it is obvious to see that the constraint is violated if and only if there

exists any pair of i and j satisfying $M_{ij} = 1$, $b_{\tilde{I}(i)} = 1$ and $b_{\tilde{G}(j)} = 1$. We want to show that Equation (2.13) is violated under the same condition.

Equation (2.13) produces $N_{\tilde{G}}$ linear inequalities. Let us consider the jth inequality:

$$\sum_{i=1}^{N_{\tilde{I}}} M_{ij} b_{\tilde{I}(i)} + N_{\tilde{I}} b_{\tilde{G}(j)} \leq N_{\tilde{I}}. \tag{2.14}$$

1. If $b_{\tilde{G}(j)} = 0$, because M_{ij} and $b_{\tilde{I}(i)}$ are both binary variables taking either 0 or 1, $\sum_{i=1}^{N_{\tilde{I}}} M_{ij} b_{\tilde{I}(i)} \leq N_{\tilde{I}}$ is always true. This suggests that regardless the choice of $\mathbf{b}_{\tilde{I}}$, the constraint in (2.14) is satisfied.

2. If $b_{\tilde{G}(j)} = 1$, $N_{\tilde{I}} b_{\tilde{G}(j)}$ equals to $N_{\tilde{I}}$. If there exits any i satisfying $M_{ij} = 1$ and $b_{\tilde{I}(i)} = 1$, then $\sum_{i=1}^{N_{\tilde{I}}} M_{ij} b_{\tilde{I}(i)}$ is larger than 0, making the inequality untrue. In order for the inequality to hold, the first term must be 0, meaning when $b_{\tilde{G}(j)} = 1$, M_{ij} and $b_{\tilde{I}(i)}$ cannot be 1 at the same time.

The above argument extends to all j's.

As such, we can replace the original constraint with the inequality in (2.13), which is linear. As the objective function is also linear, we can use efficient linear binary programming methods (such as a branch-and-bound algorithm [66]) to solve the optimization problem.

2.4 Experimental Results

2.4.1 Parameter Selection

One parameter used throughout the algorithm is the average diameter of the nanoparticle size, denoted by d_0. The d_0 can be considered as the average effect of a_0 and b_0 in the particle shape model, and it is used as the input to set a number of other settings in the algorithm. The value of d_0 in a TEM image is largely determined by a particle's actual size and the resolution level set in the TEM. Informed by our material science collaborators,

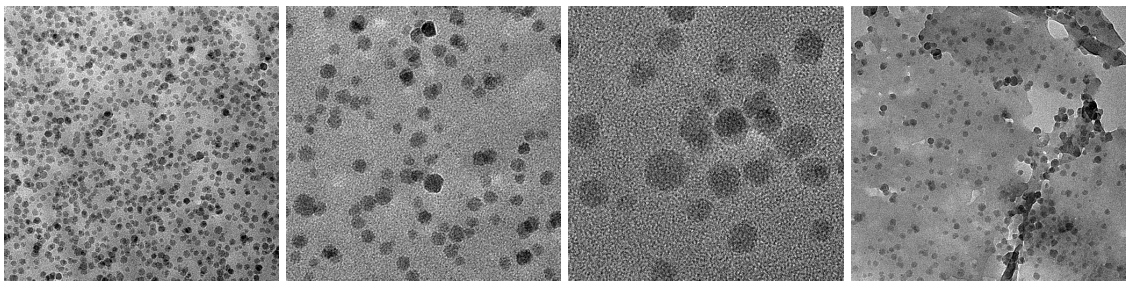
we know about the average physical diameter of the nanoparticles to be blended in the host material. The physical size is used to estimate d_0 under a specific TEM resolution, which is a good enough initial estimate and can be refined once the TEM image is processed. The refined estimate of d_0 can be used to run the whole algorithm a second time so as to improve the quality of the processing.

In the main part of the algorithm, there are two other sets of parameters: (1) r_U and r_L that are used to categorize the detection outcomes into three groups, i.e., unrelated, consensus, and conflicting; (2) λ in the fitness score. We empirically choose $r_U = 0.8$ and $r_L = 0.2$. We test many TEM images and find that these choices produce rather robust categorizations consistent with human interpretation. We want to set the pixel intensity gap to be about one-tenth of the grayness levels from the brightest to the darkest in the TEM images, in order to differentiate a particle's foreground from its surrounding background. For noisy TEM images, this gap appears reasonable. Given that our TEM images have roughly 200 grayness levels, it suggests that the gap is going to be 20, and according to Equation (2.8), this sets $\lambda = 100$.

2.4.2 TEM Images Used in the Test

We test a total of 32 TEM images taken of the bisphenol-F epoxy resin samples that are blended with silica nanoparticles. These images can be grouped into four categories. The first three categories correspond to different resolution levels of TEM. All TEM images have $1,024 \times 1,024$ pixels. So the low resolution image is taken from a big view field of about $1,000 \times 1,000$ nm; the medium resolution taken from a view field of about 500×500 nm; the high resolution image taken from the smallest view field of 250×250 nm. The last category, and also the fourth, of the images is the one having an uneven background of particular patterns. This background pattern is a result of inconsistency in the resin properties, so that the nanoparticles do not disperse well as they are blended in. This set of

images with uneven background is taken under the low resolution. Figure 2.12 shows one typical image for each of the kinds, in which Figure 2.12 (a) and (b) are the same images as those shown in Figure 2.1.



(a) Low resolution (b) Medium resolution (c) High resolution (d) Uneven background

Figure 2.12: Samples of TEM images.

As we explained earlier, the average particle size d_0 in an image is affected by the resolution of TEM. The ranges of these d_0's in the aforementioned four categories of images are presented in Table 2.1.

Table 2.1: Estimates of d_0 in the TEM images.

	Low resolution	Medium resolution	High resolution	Uneven background
d_0	20	50	$100 - 120$	20

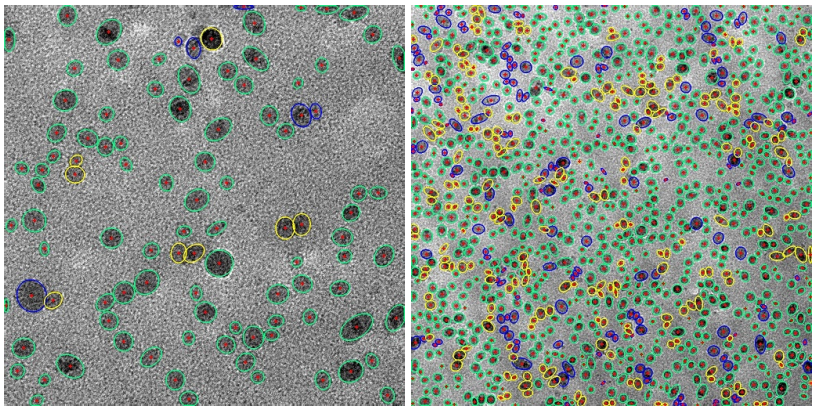
2.4.3 Comparing the Integrated Approach with Individual Pipeline of Processing

Using the two images in Figure 2.1, we want to show where the integrated approach improves upon the individual pipelines of processing. Figure 2.13 presents the detection

outcomes. The two images illustrate the results of the integrated processing. The images are color coded: green means a consensus detection, blue means that an intensity-based detection prevails, and yellow means that a gradient-based detection prevails. In the low resolution image ("F10_8"), there are 721 consensus detections, out of $1,100$ particles finally detected. Among the 379 conflicting detections, 162, or 43%, final outcomes come from the intensity-based processing, whereas 217, or 57%, come from the gradient-based processing. The respective numbers for the medium resolution image ("F3-2_7") are: 103 total particles, 85 consensus detections, 18 conflicting detections, and among those particles, 9, or 50%, are from the intensity-based processing, whereas the other 9, or 50%, from the gradient-based processing. Figure 2.14 presents the outcomes of the integrated processing for other two categories of TEM images; the same color code applies. We observe again that the integrated processing improves upon the individual pipeline of processing and we believe that this is a key advantage of the integrated approach, as it makes use of the image information fully and compensate for the limitations of the approaches emphasizing too much on one type of image information.

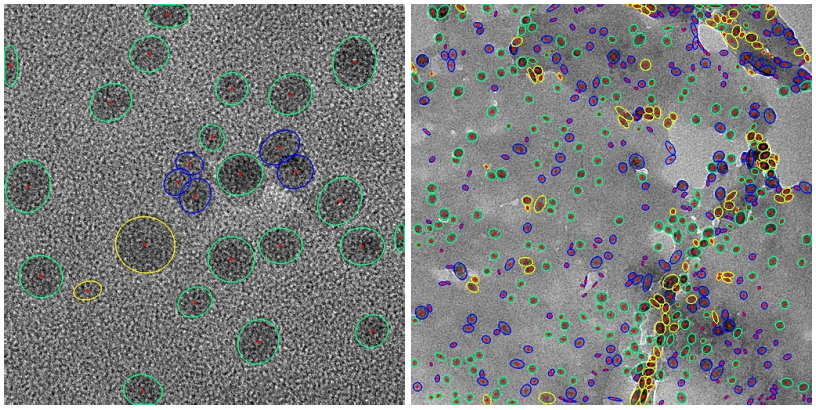
2.4.4 Test Outcomes of Four Kinds of TEM Images

To quantify the performance of our method, we run the algorithm on all 32 TEM images and report the number of particles they are able to identify. For the medium and high resolution images, we are able to manually label the particles and treat the manual outcome as our ground truth. These detection results are included in Table 2.2. In Table 2.2, for the individual pipeline of processing, we report the numbers of the total particle detections as well as the number of the consensus detections and the conflicting detection outcomes selected by the integrated method. The percentages of the conflicting detections selected from each pipeline are also shown in the table. For further comparison, we define the dissimilarity between the detected outcomes and the ground truth as the average distance



(a) Medium resolution (b) Low resolution

Figure 2.13: Comparison of individual pipelines of processing. The left image corresponds to Figure 2.1(a) (medium-resolution image), while the right image corresponds to Figure 2.1(b) (low-resolution image). Green particles are those from the consensus detections; blue particles are an intensity-based detection; yellow particles are an gradient-based detection.



(a) High resolution (b) Uneven Background

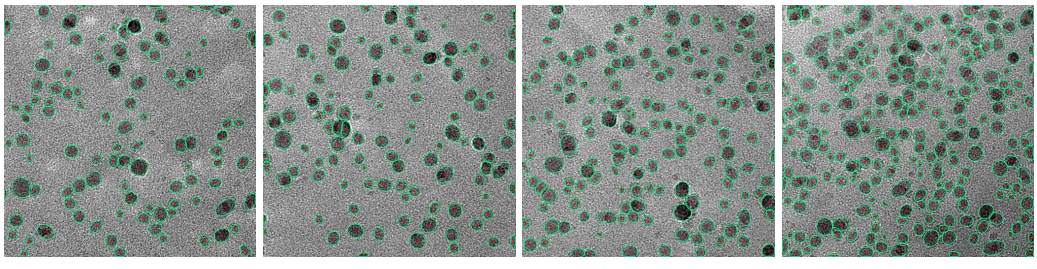
Figure 2.14: Comparison of individual pipelines of processing for other two categories. The left image corresponds to Figure “F3-2_11" (high-resolution image), whereas the right image corresponds to Figure “F8-2_16" (image with uneven background). Color coding is the same as in Figure 2.13.

between the nearest centers of different point sets, and show the boxplots of the comparison results in Figure 2.16. The smaller the dissimilarity, the better a detection outcome. Because of the availability of the results in Table 2.2 and Figure 2.16, we only present half of the processed images in Figure 2.15, as inclusion of all images makes the dissertation's file size too large.

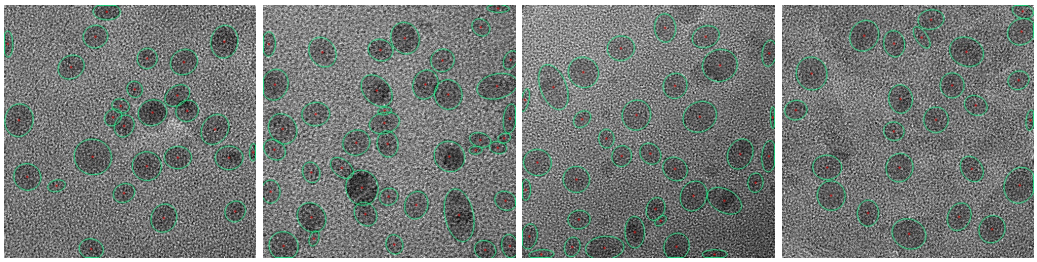
Table 2.2: Comparison of particle detections for medium and high resolution images.

TEM image	Ground truth	Intensity Based	Gradient Based	Consensus Detections	Selected From Intensity		Selected From Gradient		Integrated approach
					Number	Percentage	Number	Percentage	
Medium resolution images									
F3-2_6	103	99	97	85	9	56.3%	7	43.7%	101
F3-2_7	104	100	99	85	9	50%	9	50%	103
F3-2_8	100	99	98	73	10	35.7%	18	64.3%	101
F8-2_6	113	108	111	98	5	35.7%	9	64.3%	112
F8-2_7	134	126	131	119	1	7.7%	12	92.3%	132
F8-2_8	148	143	142	119	14	51.8%	13	48.2%	146
F10_10	214	201	195	114	64	66%	33	34%	211
F10_12	179	175	162	141	30	88.2%	4	11.8%	175
High resolution images									
F3-2_9	24	24	25	14	0	0%	10	100%	24
F3-2_10	26	26	24	11	4	28.6%	10	71.4%	25
F3-2_11	26	33	25	20	5	71.4%	2	28.6%	27
F8-2_10	42	44	37	17	12	50%	12	50%	41
F8-2_11	44	41	35	20	16	69.6%	7	30.4%	43
F10_13	37	41	36	23	4	33.3%	8	66.7%	35
F10_15	47	50	34	17	24	82.8%	5	17.2%	46
F10-2_17	25	31	25	19	4	66.7%	2	33.3%	25

The results presented in Table 2.2 and Figure 2.16 demonstrate the effectiveness of the integrated approach. Both of intensity-based and gradient-based processing contribute to the intergraded results and combining their strengths allows the proposed method to achieve a high degree of accuracy consistently across the samples. We also conduct an analysis of variance (ANOVA) [67] on the dissimilarity of three groups (integrated approaches, intensity-based only and gradient-based only) for the medium and high resolution images. For the medium resolution images, the p-value of an one-way ANOVA test is 0.0124 between the integrated approach and the intensity-based approach and 0.0013



(a) F3-2_7 (b) F8-2_6 (c) F8-2_8 (d) F10_10



(e) F3-2_11 (f) F8-2_10 (g) F10_13 (h) F10-2_17

Figure 2.15: The processed outcomes of medium-resolution (top row) images and high-resolution (bottom row) images.

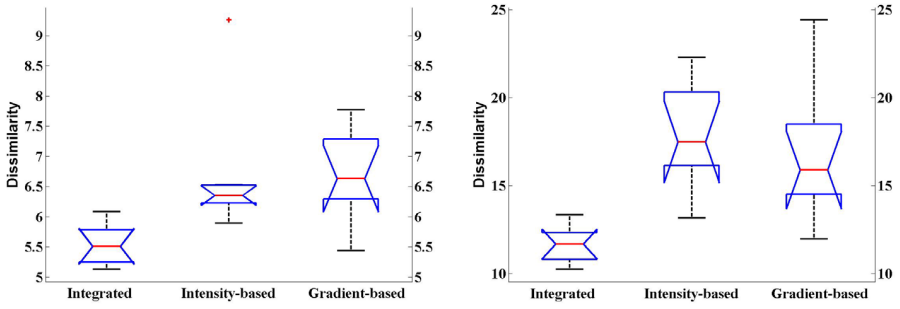


Figure 2.16: The boxplot of the dissimilarity metric for (a) medium resolution images and (b) high resolution images.

between the integrated approach and the gradient-based approach. For the high resolution images, the p-value is 0.0001 between the integrated approach and the intensity-based approach and 0.0025 between the integrated approach and the gradient-based approach.

For the low resolution images including the ones with uneven background, it is difficult to manually count and identify all the particles, as they usually have over hundreds or even thousands of particles. What we do here is to present the processed outcomes of individual images in Figure 2.17, so that people can visually sense how the method performs. We still only show half of the results due to the images large size. We present a table, similar to Table 2.2, but it does not have the ground truth column. For the intensity-based and gradient-based approaches, we again report the numbers of particles it detects and the numbers of the conflicted outcomes selected by the integrated method. Combining both Table 2.3 and Figure 2.17, we believe that the proposed method presents an advantage in achieving robust detections when the image quality varies.

Table 2.2 and Table 2.3 also suggest that the two pipelines of processing make similar contributions for the low, medium and high-resolution TEM images. But for those images with uneven background, more conflicting outcomes are selected from the intensity-based processing than from the gradient-based processing. We believe that the unevenness in background intensity causes confusion in using the gradient information, making the intensity-based processing more accurate and the gradient-based processing less so.

2.4.5 Computation Time

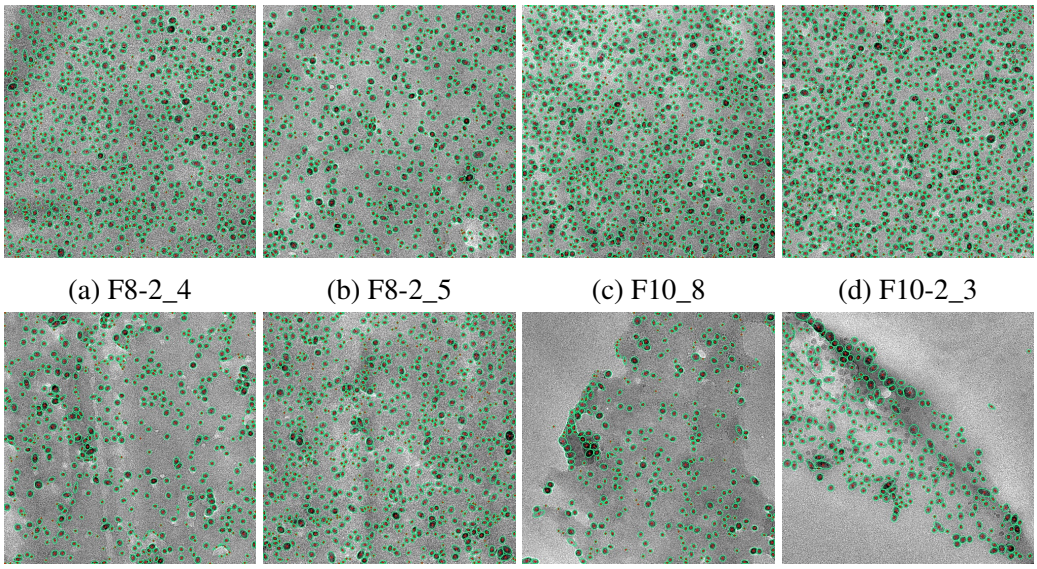
People perceive that the time spent to process a TEM image is proportional to the number of particles in an image. This turns out untrue. The processing time in fact depends heavily on the resolution level of an image; see Figure 2.18. The horizontal axis is the value of d_0 related to an image's resolution level. As the resolution gets higher, d_0 gets bigger, even for particles of the same physical size. Our algorithm spent longer time to process the

Table 2.3: Comparison of particle detections for low resolution images.

TEM image	Intensity Based	Gradient Based	Consensus Detections	Selected From Intensity		Selected From Gradient		Integrated approach
				Number	Percentage	Number	Percentage	
Low resolution images								
F3-2_16	826	695	403	257	61.6%	160	38.4%	820
F8_8	1197	997	595	425	67.9%	201	32.1%	1221
F8-2_4	871	822	575	189	62.4%	141	37.6%	878
F8-2_5	633	678	510	65	40.4%	96	59.6%	671
F10_7	885	924	667	109	43.3%	143	56.7%	919
F10_8	1041	1077	721	162	42.7%	217	57.3%	1100
F10_9	1115	1153	730	211	46.6%	242	53.4%	1183
F10-2_3	1053	1096	763	153	43.5%	199	56.3%	1115
Uneven background images								
F3-2_4	502	487	294	133	60.7%	86	39.3%	513
F3-2_5	465	463	228	150	55.3%	121	44.7%	499
F3-2_15	815	712	466	222	64.5%	122	35.5%	810
F8_13	291	200	95	124	73.8%	44	26.2%	263
F8-2_15	327	309	159	133	65.3%	60	34.7%	332
F8-2_16	556	398	199	247	73.3%	90	26.7%	536
F10-2_12	480	187	102	303	95.3%	15	4.7%	420
F10-2_13	290	259	165	80	60.6%	52	39.4%	297

high resolution images than the low-resolution ones. But the overall time is manageable. For the 32 images, the longest processing time is about 10 minutes. Recall that our method is intended to be an offline processing tool, so our material science collaborators deem 10 minutes very much acceptable.

The most time consuming part of our algorithm is the two iterative processing components: the active contour and the iterative voting. When processing the high resolution images, the heavier noise and lower contrast make it harder for the active contour method to find the optimal solution of their energy functional. So it takes a longer time to converge. When the iterative voting is applied to the high resolution images, the large diameter of particles d_0 leads to a large voting region for each step in its iteration, also causing a longer time for the method to execute.



(a) F8-2_4 (b) F8-2_5 (c) F10_8 (d) F10-2_3

(e) F3-2_4 (f) F3-2_15 (g) F8-2_15 (h) F10-2_13

Figure 2.17: The processed outcomes of low-resolution images (top row) and the images with uneven background (bottom row).

2.4.6 Parameter Sensitivity

In this section, we discuss the effect of the input parameter d_0 on the detection results. We test a given set of TEM images using different d_0's and generate the box-plot of the dissimilarity and total processing time in Figure 2.19. We choose the medium resolution TEM images because (a) these images contain a good number of nanoparticles and (b) the number of particles is manageable so that we can manually verify the ground-truth. The recommended value of d_0 is 50, which is the middle value of the parameter's range. Following the evaluation methodology suggested in [68, 69], we also fit a 3-degree polynomial of d_0 for its mean values of the dissimilarity and total time; this 3-degree polynomial is shown as the black dashed line in the respective plots.

Figure 2.19 shows that d_0 does play an importance role in affecting detection quality as well as detection time. If d_0 is chosen too small or too big, both detection quality and

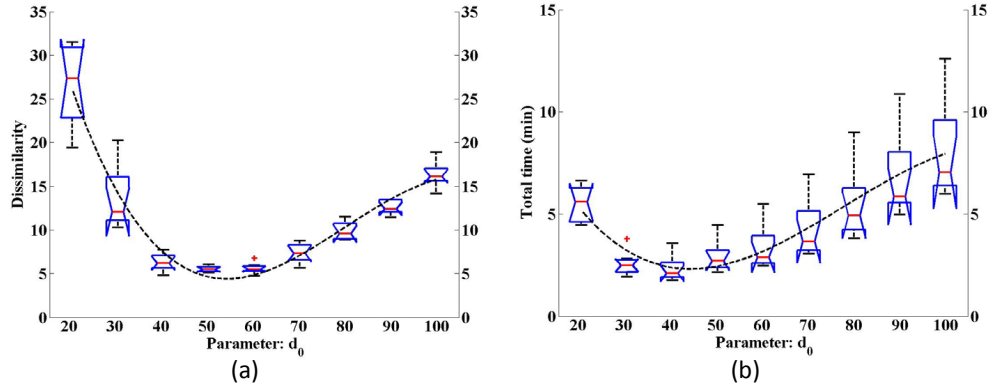


Figure 2.18: Computational time of the algorithm. The horizontal axis is d_0, and the vertical axis is the processing time in minutes.

processing time will be adversely affected. We also observe that underestimation of d_0 harms the detection quality more than overestimation, while overestimation prolongs the processing time more. Nevertheless, both detection quality and processing time remain reasonably stable when d_0 is chosen between 40 and 60, namely within 20% deviation of the nominal particle size. This range of allowance makes it practical to use a rough estimate of the particle diameter in the proposed method to produce robust detection results.

2.5 Summary

In this chapter, we proposed a new method to detect the nanoparticles in noisy (TEM) images. The main contribution of the work is that we present a framework leading to robust processing capability. This framework entails two pipelines of processing in parallel, making use of complementary image information, followed by a binary integer optimization procedure to resolve detection conflicts and select better outcomes. Our method can solve the particle detection problem for TEM images with low contrast and heavy noise, making the new method particularly useful in the application of non-metallic nano material analysis.

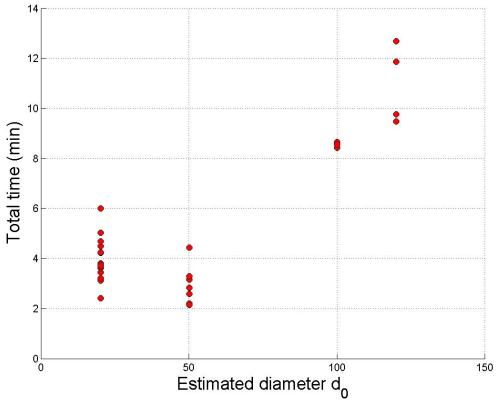


Figure 2.19: The box-plot and response curve of (a) dissimilarity and (b) total processing time, with respect to d_0, for the medium resolution TEM images.

We do want to point out a few possible extensions of our work. When in the future a new pipeline of processing is discovered to complement the existing two processing pipelines, our BIP formulation does allow an extension to include those. What needs to be done is to amend the constraint conditions to incorporate more than two detection outcomes and devise a conflict matrix $\mathbf{M}$, making sure that still only one outcome is chosen eventually. Application front, a possible extension of the current work is to explore the method's applicability to bio-image processing like cell detection or object detection from satellite images especially when the quality of those images is low.

3. IDENTIFYING MULTI-STAGE NANOCRYSTAL GROWTH USING *IN SITU* TEM VIDEO DATA[1]

Under a controlled circumstance, atoms, ions and molecules can simultaneously grow into nanoparticles with a highly ordered structure, and such nanoparticles are often referred to as nanocrystals. *In situ* TEM technique has caught recent attentions in material science research because the *in situ* technology provides the capability of directly observing the growth process of nanocrystals, and makes possible discoveries that *ex situ* instruments cannot. As more and more dynamic TEM video data becomes available, one of the bottlenecks appears to be the lack of automated, quantitative and dynamic analytic tools that can process the video data efficiently. The current processing is largely manual in nature and laborious, while most of the automatic tools only focus on static TEM images. The absence of the automated processing of TEM videos does not come as a surprise, as the growth of nanocrystals is highly stochastic and goes through multiple stages. We introduce a method in Chapter 3, suitable for analyzing the *in situ* TEM videos in an automated and effective way. The method learns and tracks the normalized particle size distribution and identifies the phase change points delineating the stages in nanocrystal growth. Using the outcome of the change point detection process, we produce a hybrid multi-stage growth model and test it on an *in situ* TEM video, made available in 2009 by *Science*.

3.1 Introduction

In situ TEM is a promising new technology available to scientists for making discoveries in the nanoscale world. *In situ* TEM uses a special sample holder, allowing motion pictures to be taken while the nano-objects in the sample holder are initiating, crystalizing and

[1]Reprinted with permission from Y. Qian, J. Z. Huang, and Y. Ding, "Identifying multi-stage nanocrystal growth using *in situ* TEM video data," *IISE Transactions*, vol. 49, no. 5, pp. 532–543, 2017. https://doi.org/10.1080/24725854.2016.1251666, Copyright © 2017 by Taylor & Francis.

morphing into different sizes and shapes. The unique capability of *in situ* TEM is that it captures the dynamic changes at the nano or sub-nano resolution and provides the opportunity of studying, and the potential of understanding, the mechanisms of multistage growth of nanocrystals. Material scientists point out that understanding and modeling the growth trajectory of nanocrystal are the important first step leading to the control of nanocrystal synthesis processes in the long run, and expediting the discoveries of how a new nanomaterial works [26, 27]. Two *in situ* TEM video segments of a platinum nanocrystal growth were made available by Zheng et al. [1] as parts of the supplementary material to their publication. The short segment is about 21.2 seconds in duration with 30 frames per second, and the long segment is 76.6 seconds with 15 frames per second. In addition to [1], a number of other researchers also reported the use of *in situ* TEM videos in their study of the mechanism of nanocrystal growth [2, 70, 71, 72, 73].

The current practice of processing the *in situ* TEM videos is largely manual in nature, working typically as follows. Researchers label individual particles in each time frame of the video, measure the sizes of particles, count the number and categorize their shapes, then plot particle size/shape related histograms or report relevant statistics that may lead to some insights into nanocrystal growth. Image processing software is used to facilitate the isolation of overlapped nanocrystals and the measurement of their sizes or aspect ratios (the ratio between long and short axes). One popular tool of this kind is the freeware `ImageJ` [6], developed by the National Institute of Health, which was used, for instance, in the work of [73]. There are also many recent works [7, 9, 74] improving the accuracy of detecting nanoparticles in TEM images significantly. However, those processing tools used can only handle static pictures one frame at a time and do not have the ability to extract dynamic information from the videos.

The manual processing appears to be a bottleneck preventing scientists from taking full advantage of the capability enabled by the new microscopy technique. Processing video

data, considering their sheer volumes and data sizes, is laborious and time-consuming. Processing multiple clips of videos is also repetitive and prone to human errors.

More importantly, one crucial limitation of manual operation is the difficulty in identifying the change points in a nanocrystal growth trajectory going through multiple phases. It is nearly impossible for a person to identify change points accurately by simply looking at the videos, while a nanocrystal growth going through multiple stages is common. Indeed, past experiments have shown that a nanocrystal growth can be driven by different kinetics [75, 76] in various stages. Researchers have developed mathematical models for two kinds of growth mechanisms: the traditional monomer attachment growth, also known as Ostwald ripening [77, 78], and the non-classical mechanism, like the orientated attachment [79, 80]. To take advantage of these models for describing the dynamics of nanocrystal growth, a data analytic tool is pressingly needed for processing the *in situ* TEM videos and detecting the phase change points delineating the growth stages.

The lack of an automated tool fulfilling the aforementioned tasks does not come as a surprise, as the nanocrystal growth trajectory is highly stochastic. The current practice, manual in nature aside, primarily uses some simple size/shape statistics, such as the sample average, to represent the nanocrystal evolution. However, these simple statistics are not sufficient in fully summarizing the information in the TEM video data. In recent years, some researchers have made the first step in going beyond simple summary statistics. Park [81] learned the multiple-path growth trajectory of nano-crystals from the *in situ* or *ex situ* TEM images. Park et al. [16] proposed a method to track the interacting nanocrystals through the growth process in an *in situ* TEM video. This line of work focuses on tracking an individual nanoparticle growing through various stages. Woehl et al. [2] proposed to identify the growth mechanism with the normalized particle size distribution (NPSD). They estimated the NPSD by collecting nanocrystal size information from TEM videos. Since the nanocrystal size measurements from all time frames were pooled together in

their work to obtain a single static NPSD, Woehl et al. [2] did not describe the dynamic change underlying the nanocrystal growth.

In this work, we propose a method to estimate the time-varying NPSD, i.e., one NPSD at each time frame, using images from *in situ* TEM video. For each time frame, one could fit a probability density function for the normalized nanocrystal sizes, using a standard probability density estimation method, such as the penalized B-spline method [18]. However, direct application of standard methods does not give good density estimation due to small sample sizes – there are too few nanocrystals at each time frame. To overcome the small sample size problem, we propose to extend the penalized B-spline density estimation in the following sense. In the usual penalized B-spline formulation [18], the log density function is modeled as a linear combination of B-spline basis functions, and the penalized likelihood method is used to estimate the coefficients of the B-spline expansion. In our extended formulation, the log likelihoods from all time frames are added together and, in addition to the penalty that ensures smoothness of each estimated density function, another penalty term is included to guarantee that the time-varying density functions change smoothly over time. This new formulation of penalized B-splines allows us to borrow information across time frames to obtain more reliable density estimation.

Under some fixed growth mechanisms, material scientists can use the self-similar analytic models to describe the theoretical NPSD [82, 83, 84], which assumes that the NPSD can be approximated by the asymptotic solution at the infinite time. Based on that assumption, after the time-varying NPSD is estimated, we can apply a change point detection method to the estimated density functions, to identify the time points of potential phase changes.

In order to facilitate the detection, we discretize each density function into a vector, and then apply the principal component analysis (PCA) [85] to represent the time-varying NPSD with a small number of principal component (PC) scores. After that, state-of-

the-art multiple change detection methods, recently developed by Killick et al. [86] and Fryzlewicz [87], could be used to detect the change points. One problem, however, is that these methods tend to detect more change points than waht the physical understanding can explain. To address this problem, we propose a selecting procedure to choose the significant change points from the candidates identified by the existing methods, using the sum of squared errors (SSE) as the criterion. We stop the process when the deduction rate of SSE is smaller than a threshold. We find that this selecting procedure yields a change point detection result that can be explained by the underlying nanocrystal growth mechanisms. In addition to NPSD, we also apply our method to the median particle size to supplement the NPSD-based change point detection.

With the change points detected using either the NPSD or the median particle size, we are able to partition the particle growth process into several stages, each of which is then described by an existing nanocrystal growth model. We applied this strategy to a published TEM video segment to build a hybrid model for the whole stage of nanocrystal growth. Using this new model, we can estimate stage-specific parameters and perform quantitative comparisons of different stages. In a comparison with the single-stage model used by Woehl et al. [2], our hybrid model is shown to be able to describe the nanocrystal growth trajectory more accurately.

The rest of this chapter is organized as follows. In Section 3.2, we briefly discuss the image preprocessing step and then introduce some definitions and notations. In Section 3.3, we present the details for modeling the time-varying NPSD. In Section 3.4, we present our change point detection approach. In Section 3.5, we conduct a sensitivity analysis of the tuning parameter used in our detection. In Section 3.6, we combine the two mechanistic models, forming a hybrid model for the whole growth stage. Several comparisons are conducted in this section. Finally in Section 3.7, we conclude our work.

3.2 Image Preprocessing and Notations

We describe our methodology using the long segment video provided by Zheng et al. [1] (file name: '1172104s1.mov' in their supplementary material). We select the long segment because its duration is long enough to contain multiple growth stages. Although using the specific video as an example, the development of our stage identification and change point detection method is not tailored to this particular example. We believe our methodology from this section can be readily applied to other *in situ* TEM videos of nanocrystal growth.

3.2.1 Video Preprocessing

Before identifying the nanocrystal growth, the first step is to detect nanocrystals in the image of each video frame and extract their morphology information. One particular emphasis is to address the issue of image segmentation among the aggregated nanoparticles. To fulfill this preprocessing task, we used an image processing method developed by our own team [88], which is particularly potent of handling low-contrast and noisy TEM images and performs better than other methods for handling aggregated nanoparticles [9, 7, 11]. The detection results at some selected time points are shown in Figure 3.1.

In this study, as in the original paper [1], we are primarily concerned with the change in nanocrystal size, as the shapes of the nanocrystals are rather uniform. The nanocrystal size is, understandably, characterized by its radius. Denote by r_{st} the radius of the sth particle at time t, and by $\bar{r}_t$ the mean radius and $\tilde{r}_t$ the median radius, both at time t. Same as in [1] and [2], the radius r_{st} is defined as $\sqrt{P_{st}/\pi}$, where P_{st} is the area of the corresponding particle.

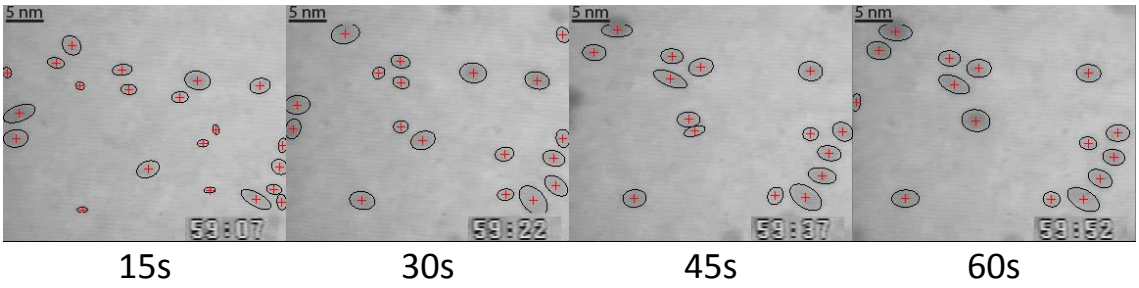


Figure 3.1: Four frames from the long segment video provided by [1] and the nanocrystal detection results. The contour line indicates a nanocrystal's edge, while the '+' indicates a nanocrystal's center.

3.2.2 Definition of Normalized Particle Size Distribution

Let $G_t(r)$ denote the particle size distribution at time t. The mean radius $\bar{r}_t$ can then be expressed as:

$$\bar{r}_t = \int_0^\infty r G_t(r) dr. \tag{3.1}$$

We normalize the nanocrystal size r_{st} at time t by $\bar{r}_t$ to obtain $\phi_{st} = r_{st}/\bar{r}_t$. The normalized particle size distribution, denoted as $F_t(\phi)$, where $\phi = r/\bar{r}_t$, is the distribution of ϕ_{st} at time t. It is easy to see that $F_t(\phi)$ is determined by $G_t(r)$ and $\bar{r}_t$ as:

$$F_t(\phi) = \bar{r}_t G_t(\bar{r}_t \phi). \tag{3.2}$$

Note that both $G_t(\cdot)$ and $F_t(\cdot)$ are time-varying functions, as signified by the subscript t. NPSD provides a better measure of nanocrystal growth mechanism than the particle size distribution (PSD). Past research [2, 77, 82] has shown that when the underlying growth mechanism remains the same (within a single stage), the NPSD stays stable while the PSD always changes with the growing sizes of the nanocrystals. So a change in NPSD can be a strong clue to signify a new growth mechanism.

3.3 Penalized B-splines for Estimating NPSD

We start off with introducing the estimation of a single probability density function $F_t(\phi)$ from ϕ_{st} at time t using the method of [18]. The basic idea is to model the log density function as a linear combination of B-spline basis functions, then estimate the spline coefficients from the histogram of the observations by maximizing the penalized likelihood. Specifically, the log density can be modelled as:

$$\log(F(\phi)) = \sum_{j=1}^{n} a_{jt} B_j(\phi) - C_t, \forall i = 1, \ldots, m, \tag{3.3}$$

where $B_j(\phi)$ is the jth B-spline basis function, n is the number of basis functions, and

$$C_t = \int_0^{\infty} \sum_{j=1}^{n} a_{jt} B_j(\phi) \mathrm{d}\phi \tag{3.4}$$

is the normalized constant. Following Eilers and Marx [18], we create a histogram by dividing the ϕ axis into m intervals to estimate the spline coefficients (in a B-spline, m is the number of knots). Denote the midpoints as ϕ_i, $i = 1, \ldots, m$. Then the B-spline function in Equation (3.3) evaluated at ϕ_i can be written as:

$$\eta_{it} = \sum_{j=1}^{n} a_{jt} B_j(\phi_i), \forall i = 1, \ldots, m. \tag{3.5}$$

The number of observations falling in the i-th interval at the time frame t, denoted by y_{it}, can be assumed as Poisson distributed with density $\exp(\eta_{it})$. The penalized Poisson likelihood function of $\{a_{jt}\}$ is:

$$L_t(\{a_{jt}\}) = \sum_{i=1}^{m} y_{it}\eta_{it} - \sum_{i=1}^{m} \exp(\eta_{it}) - \lambda_1 \sum_{j=1}^{n-1} \frac{(\Delta_1 a_{jt})^2}{2}, \tag{3.6}$$

where Δ_1 is a difference operator with $\Delta_1 a_{jt} = a_{(j+1)t} - a_{jt}$. In the above objective function, the first and second terms correspond to the Poisson likelihood, measuring the goodness-of-fit of $F_t(\phi)$ to the histogram $\{y_{1t}, \cdots, y_{mt}\}$; the third term is the roughness penalty, with λ_1 being the penalty parameter, ensuring smoothness of the estimated density. One should maximize $L_t(\{a_{jt}\})$ and then substitute the maximizer $\{\hat{a}_{jt}\}$ to Equation (3.3) to obtain the estimated probability density $\hat{F}_t(\phi)$ for a single time frame.

When the nanocrystals are too few at some time frames, estimating density functions separately at each time frame does not produce good results. In our revised penalized B-spline formulation, we estimate the density functions by pooling all time frames data together. But unlike [2] in which the resulting NPSD is a constant function over the whole growth trajectory, we allow our NPSD to be time varying, in order to capture the growth dynamics. For this reason, we introduce an additional roughness penalty to ensure that the density functions vary smoothly over time. The new objective function is:

$$L(\{a_{jt}\}) = \sum_{t=1}^{T} L_t(\{a_{jt}\}) - \lambda_2 \sum_{j=1}^{n} \sum_{t=1}^{T-1} \frac{(\Delta_2 a_{jt})^2}{2}, \tag{3.7}$$

where Δ_2 is a difference operator with $\Delta_2 a_{jt} = a_{j(t+1)} - a_{jt}$. The λ_2 is the temporal roughness penalty parameter. This new formulation enables borrowing information among different time frames and thus improves estimation efficiency, especially at those time frames with too few nanocrystals.

We maximize the penalized log likelihood given in Equation (3.7) to obtain the spline coefficients associated with all density functions over the whole growth duration. Apparently, the algorithm developed by Eilers and Marx [18] does not apply since the new formulation has an extra index t and an extra penalty term. The main challenge is caused by the newly introduced, second penalty term, which makes the objective function not separable with respect to t.

We propose to apply the alternating direction multiplier method (ADMM) [24] to decouple the the relationship along the t index. Specifically, we replace a_{jt}'s in the second penalty term by a set of new variables z_{jt}'s and solve the optimization problem under the constraints $a_{jt} = z_{jt}$. We perform the constrained optimization by considering the augmented Lagrangian as follows:

$$\begin{aligned} L_\rho(\{a_{jt}\}, \{z_{jt}\}, \{c_{it}\}) = \sum_{t=1}^{T} L_t(\{a_{jt}\}) - \lambda_2 \sum_{j=1}^{n} \sum_{t=1}^{T-1} \frac{(\Delta_2 z_{jt})^2}{2} \\ - \rho \sum_{t=1}^{T} \sum_{j=1}^{n} c_{jt}(a_{jt} - z_{jt}) - \frac{\rho}{2} \sum_{t=1}^{T} \sum_{j=1}^{n} (a_{jt} - z_{jt})^2, \end{aligned} \tag{3.8}$$

where c_{jt}'s are the Lagrangian multipliers and ρ is the penalty parameter of the augmented Lagrangian.

Then the ADMM algorithm targets to find the saddle point of Equation (3.8), defined as:

$$(\{\hat{a}_{jt}\}, \{\hat{z}_{jt}\}, \{\hat{c}_{it}\}) = \arg \min_{\{c_{it}\}} \max_{\{a_{jt}\}, \{z_{jt}\}} L_\rho, \tag{3.9}$$

where $\{\hat{a}_{jt}\}$ will be the maximizer of the penalized log likelihood of the density functions. The saddle point is found by using the coordinate descent method [89]. The idea of the method is as follows. In the qth iteration of updating $\{a_{jt}\}$, $\{z_{jt}\}$ and $\{c_{jt}\}$, firstly we apply Eliers and Marx's algorithm to find the optimal $\{a_{jt}\}$, given $\{c_{jt}\}$ and $\{z_{jt}\}$ at their current values; then, fixing $\{a_{jt}\}$ and $\{c_{jt}\}$, the Lagrangian is a quadratic form in $\{z_{jt}\}$, whose optimization has a closed-form solution; at last the Lagrange multipliers $\{c_{jt}\}$ are updated by a "price update" step:

$$c_{jt}^{(q+1)} = c_{jt}^{(q)} + (a_{jt}^{(q)} - z_{jt}^{(q)}) \tag{3.10}$$

We continue the iteration until all those variables converge. At the convergence of the

algorithm, we substitute the convergent values of $\{a_{jt}\}$ to Equation (3.3) to get the estimated NPSD $\hat{F}_t(\phi)$ for all the time frames. The detailed steps of the ADMM algorithm are included in the Appendix A.1. We also list the steps of the algorithm in Algorithm 1.

To estimate the NPSD using the video taken by Zheng et al. [1], we set $n = 10$ (the number of B-spline basis), $m = 50$ (the number of knots), and $T = 1,148$ (the number of frames in the video). We choose the order B-spline as 2. The estimation is robust with respect to those parameters, so we can choose any reasonable values. The parameter ρ only affects the convergence speed of ADMM, so that as long as the algorithm converges, there is no need to tune it. We set it as 9.0 in this application. The remaining tuning parameters λ_1 and λ_2 can be set by the Akaike information criterion (AIC) like in [18] as

$$\text{AIC}(\lambda_1, \lambda_2) = \text{dev}(\lambda_1, \lambda_2) + 2\text{dim}(\lambda_1, \lambda_2), \tag{3.11}$$

where $\text{dev}(\lambda_1, \lambda_2)$ is the deviance of the estimated curves, and $\text{dim}(\lambda_1, \lambda_2)$ is the effective dimension of parameters. The deviance is defined as:

$$\text{dev}(\lambda_1, \lambda_2) = 2\sum_{t=1}^{T}\sum_{i=1}^{m} y_{it} \ln y_{it} - 2\sum_{t=1}^{T}\sum_{j=1}^{n} \hat{a}_{jt} B_{jt}^{+}. \tag{3.12}$$

And we define the effective dimension of parameters as:

$$\text{dim}(\lambda_1, \lambda_2) = \text{tr}\{(B'B + \lambda_1 D_1' D_1)^{-1} B'B\}\text{tr}\{(I_T + \lambda_2 D_2 D_2')^{-1}\}, \tag{3.13}$$

where $\text{tr}\{\cdot\}$ is the trace of the corresponding matrix. By minimizing $\text{AIC}(\lambda_1, \lambda_2)$, the two tuning parameters are chosen as $\lambda_1 = 0.5$, $\lambda_2 = 1.5$.

We show in Figure 3.2 the NPSDs estimated at $t = 10$s, 40s and 70s. The same approach can also be used to estimate PSD, by replacing the observations ϕ_{st} with r_{st} and replacing knots ϕ_i with r_i. The parameters m, n, T, λ_1, λ_2 and ρ are set the same as those

Algorithm 1 Detailed algorithm for solving the revised penalized B-spline density estimation

1. Set tuning parameters m, n, λ_1, λ_2 and ρ.

2. Construct the B-spline basis function $B_j(\phi_i)$ according to the knots ϕ_i for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Then we calculate $B_t^+ = [B_{1t}^+, \cdots, B_{nt}^+]'$ and $B_{jt}^+ = \sum_s B_j(\phi_{st})$.

3. Initialize $A^{(0)}$, $Z^{(0)}$ and $C^{(0)}$. We recommend setting $(H^{(0)})_{it} = \log(y_{it})$, $A^{(0)} = Z^{(0)} = B^{-1}H^{(0)}$ and $C^{(0)} = \mathbf{0}$.

4. Set $q = 0$.

5. Update $A^{(q+1)}$. For $t = 1, \ldots, T$, we update each column as follows:

   (a) Set $\hat{a}_t = a_t^{(q)}$.

   (b) Solve the following equation:

   $$\begin{aligned} & B_t^+ - B' \exp(B\hat{a}_t) + B'B\hat{a}_t + \rho(z_t^{(q)} - c_t^{(q)}) \\ = & [B'B + \lambda_1 D_1'D_1 + \rho I_n]a_t, \end{aligned} \tag{3.14}$$

   where $\hat{a}_t$ is the result estimated from the previous iteration, and D_1 is an $n \times n$ matrix with $(D_1)_{jj}$ as -1, $(D_1)_{j(j-1)}$ as 1, for $j = 2, \ldots, n$, and all other elements as 0.

   Use the solution of a_t to update $\hat{a}_t$.

   (c) Repeat the previous step until $\hat{a}_t$ converges, then let $a_t^{(q+1)} = \hat{a}_t$.

6. Update $Z^{(q+1)}$ by solving the following equation:

   $$Z^{(q+1)} = \left[A^{(q+1)} + C^{(q)}\right](I_T + \frac{\lambda_2}{\rho} D_2 D_2')^{(-1)}, \tag{3.15}$$

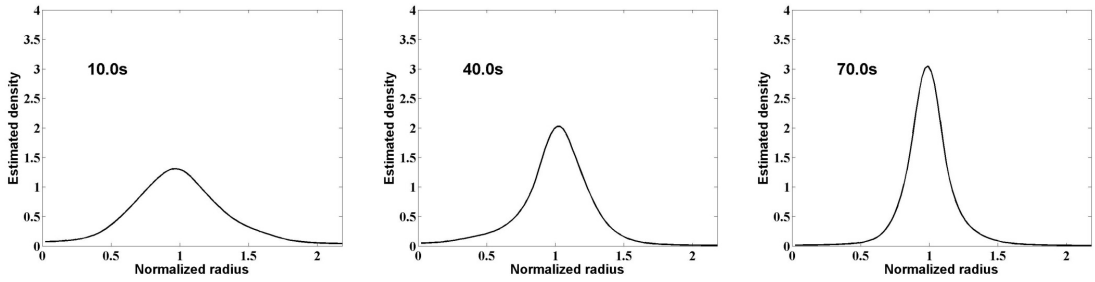
   where D_2 is a $T \times T$ matrix with $(D_2)_{tt}$ as -1, $(D_2)_{t(t+1)}$ as 1, for $t = 1, \ldots, T-1$, and all other elements as 0.

7. Update $C^{(q+1)}$ via the following equation:

   $$C^{(q+1)} = C^{(q)} + (A^{(q+1)} - Z^{(q+1)}), \tag{3.16}$$

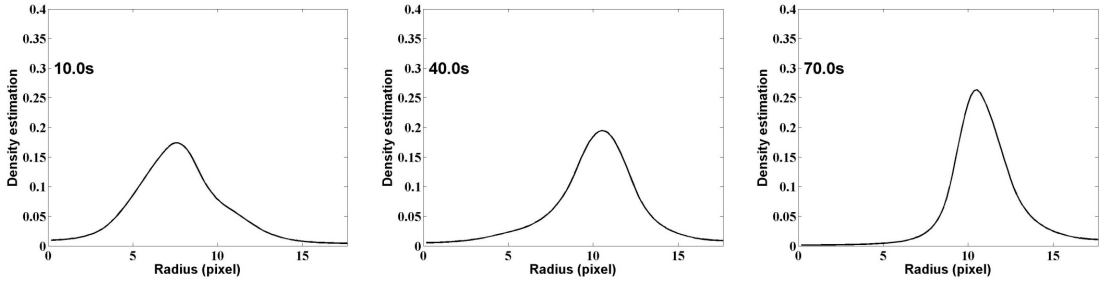
   then let $q = q + 1$.

8. Repeat Step 5 to 7 until A, Z and C all converge.



(a) NPSD at 10s (b) NPSD at 40s (c) NPSD at 70s

Figure 3.2: The estimated NPSDs at 10s, 40s and 70s.



(a) PSD at 10s (b) PSD at 40s (c) PSD at 70s

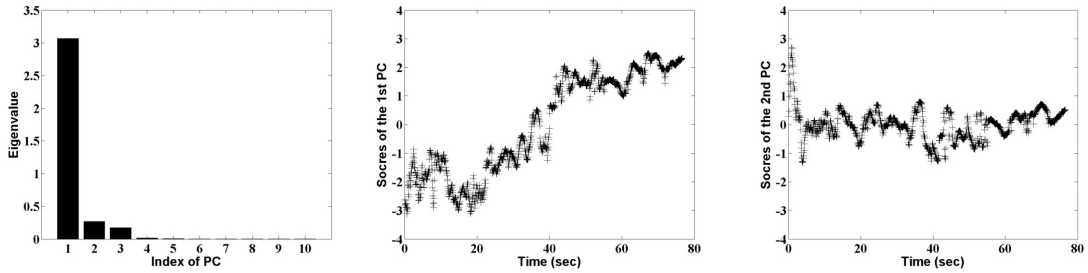
Figure 3.3: The estimated PSDs at 10s, 40s and 70s.

used in the estimation of NPSD. The PSDs estimated at $t = 10$s, 40s and 70s are shown in Figure 3.3.

3.4 Change Point Detection

The estimated NPSD $\hat{F}_t(\phi)$ is available to us as a vector at each t, i.e., $\{\hat{F}_t(\phi_1), \ldots, \hat{F}_t(\phi_m)\}$. To detect a change point in $\hat{F}_t(\phi)$ amounts to a multivariate detection problem, and a common strategy to make such detection effective is to reduce the dimension of the vector by using the principal component analysis (PCA) [85]. PCA attempts to find a small number of significant projections of the original vector onto a lower-dimensional space, which is supposed to well-represent the original vector. When applying PCA to our NPSD, it turns out that only the first principal component (PC) is significant. In Figure 3.4, we plot the first 10 eigenvalues corresponding to the respective principal components, as well as the scores of the 1st and 2nd PCs. The eigenvalue of the 1st PC is much larger than that of the other PCs. In fact, the 1st PC explains 86.5% of the total variance of the original data. In addition to considering the numerical percentage of the 1st PC, we observe that its score exhibits a clear pattern, while that of the 2nd PC appears random, reassuring the decision to use the 1st PC only for our detection purpose. So in the sequel, we work with the scores of the first PC, which is denoted by $\hat{p}_t$. But we do want to note that not always is only the first PC significant. In the case that the significant PCs are more than one, we would apply a multivariate change point detection framework, like methods in [90], on the scores of significant PCs.

Without knowing the exact number of possible change points in the process, a popular treatment, known as the binary segmentation process (BSP) [91], is to detect the most significant change point first and then continue applying the same detection method to the subsequences before and after the detected change point. The dominating criterion used in the existing BSP methods to decide the existence of a change point is the Bayesian infor-



(a) The eigenvalues corresponding to each PC

(b) Scores of the 1st PC of NPSD

(c) Scores of the 2nd PC of NPSD

Figure 3.4: PCA of the NPSD: (a) the eigenvalues corresponding to the first ten PCs; (b) the scores of the first PC; (c) the scores of the second PC.

mation criterion, also known as the Bayesian information criterion (BIC) [92]. However, if we adopt the BSP method with BIC as the stopping criterion to our data, it will find more than 400 change points, obviously over-segmenting the nanocrystal growth trajectory. We also try some state-of-the-art multiple change point detection methods, such as the pruned exact linear time (PELT) [86] and the wild binary segmentation (WBS) [87] but they still return more change points than that the mechanisms can explain (8 change points when using PELT and 49 when using WBS).

Apparently, we need to reduce the number of change points to be consistent with the physical understanding. In doing so, we find that a robust criterion to select the change points with our data is the reduction rate in the sum of squared errors (SSE) of the piecewise constant model before and after a change point is added. Recall that NPSD is supposed to stay stable within each growth stage so that the scores of NPSD's principal components should fluctuate around a constant within a growth stage. If all the change points are correctly identified, the piecewise constant model for fitting the scores of the NPSD's principal component should produce the lowest SSE.

Given all the candidates of change points detected by one of the popular methods

(we choose PELT here, as it turns the fewest change points among all methods we have explored), we start with a constant model and then test each of those candidates. We pick the first potential change point to be the place where the largest reduction of SSE can be achieved by the two-piece constant model. If the reduction of SSE is large enough, we believe this change point is genuine and will continue the selection process. Then, we visit all the remaining candidates to find the next change point which gives the largest reduction rate of SSE. We repeat the same step until the reduction of SSE is no longer significant, i.e., it is most likely due to random noise rather than a substantial change in the process. The detailed steps are described as follows.

Suppose we have already found $c-1$ change points, denoted as $\hat{t}_1, \ldots, \hat{t}_{c-1}$, while there are g remaining candidates, denoted as $\tilde{t}_1, \ldots, \tilde{t}_g$. The next possible change point chosen from $\tilde{t}_1, \ldots, \tilde{t}_g$ is denoted as t_c. They together segment the whole data sequence into $c+1$ subsequences, denoted by $S_e, e = \{1, \ldots, c+1\}$. The overall SSE of the piecewise constant model fitting of $\hat{p}_t$ is computed as:

$$V(\hat{t}_1, \ldots, \hat{t}_{c-1}, t_c) = \sum_{e=1}^{c+1} \sum_{t \in S_e} (\hat{p}_t - b_0^{(e)})^2, \tag{3.17}$$

where $b_0^{(e)}$ is the mean of $\hat{p}_t$ within S_e. The position of the next potential change point is determined by:

$$\hat{t}_c = \arg \min_{t_c \in (\tilde{t}_1, \ldots, \tilde{t}_g)} V(\hat{t}_1, \ldots, \hat{t}_{c-1}, t_c). \tag{3.18}$$

Then we will delete $\hat{t}_c$ from the candidates $\{\tilde{t}_1, \ldots, \tilde{t}_g\}$ and continue the selecting process, until there is no remaining change point candidates.

By applying Equation (3.17) to the *in situ* TEM data of our example, we can find a series of potential change points from the 8 candidates detected by PELT, which are shown in Figure 3.5(a). Figure 3.5(b) presents the profile of the SSE, $V(\hat{t}_1, \ldots, \hat{t}_c)$, in

which $\hat{t}_{(1)}, \ldots, \hat{t}_{(8)}$ represents the order of the selection. We deem a potential change point $\hat{t}_c$ a genuine change point if the deduction rate of the SSE is larger than a threshold θ:

$$\frac{V(\hat{t}_1, \ldots, \hat{t}_{c-1}) - V(\hat{t}_1, \ldots, \hat{t}_c)}{V(\hat{t}_1, \ldots, \hat{t}_{c-1})} > \theta, \tag{3.19}$$

In other words, if including $\hat{t}_c$ reduces the SSE by more than $\theta \times 100\%$, we tend to believe that the change point is due to true process change rather than random noise. Then we continue the selection for the next potential change point. If the criterion in Equation (3.19) is not satisfied, we consider that all the significant change points have been found and stop the process.

We want to note that the PC scores is auto correlated because of the temporal penalty added in our density estimation step. Should the PC scores is severe and causing too many false alarms, the autocorrelation in the PC scores may need to be removed first before a change point detection method is applied. We recommend using a model free approach such as the unweighted batch mean [93].

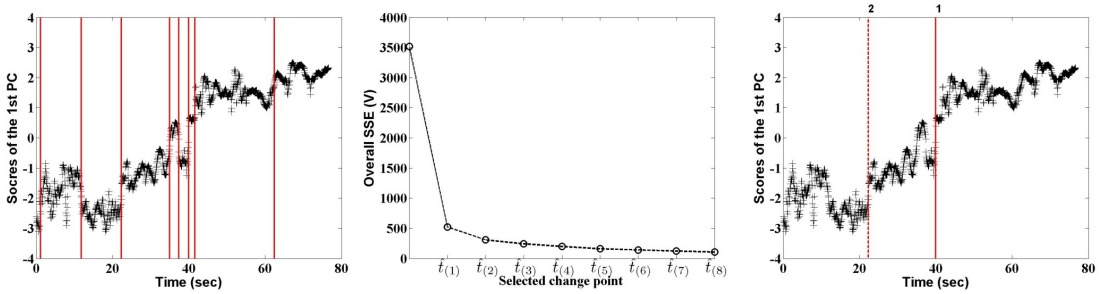


Figure 3.5: Results of the proposed change point detection using size distribution: (a) 8 potential change point candidates detected by PELT; (b) change in $V(\cdot)$ when selecting a change point at a time; (c) all change points detected when θ is varied in the range of (0.2, 0.8).

The same strategy can also be applied to a simple statistic, such as the median or mean particle size, which has more often been used to describe a nanocrystal growth due to its simplicity. As the median radius is less sensitive to outliers, we apply our method to the median particle size $\tilde{r}_t$, instead of the mean particle size $\bar{r}_t$. Unlike the NPSD, which remains relatively stable without a change point, so that its PC fluctuates around a constant, $\tilde{r}_t$ exhibits an increasing trend along the growth process.

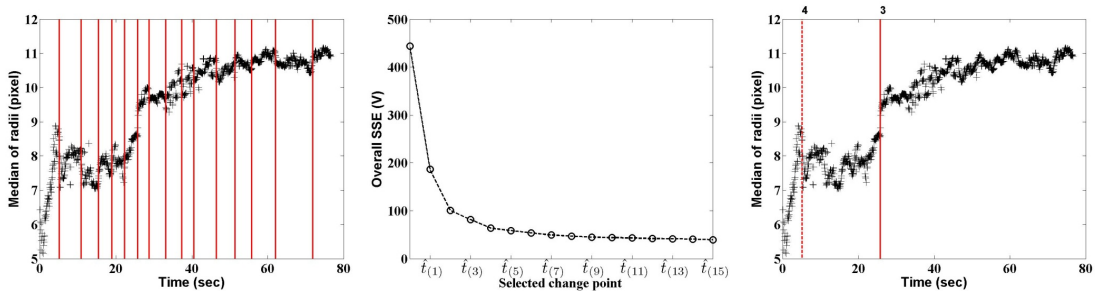
What we need to do is to revise the detection process to handle the trend. We adopt the `strucchange` package [94], which detects change points after a regression, and using it finds 15 change point candidates. To select the significant change points, we first apply a de-trending operation before performing the change point detection. Following Chen and Gupta [95], we use a linear model to de-trend the median particle size. Hence, we revise the SSE by using the residuals after fitting a piecewise linear trend model, as follows:

$$V(\hat{t}_1, \ldots, \hat{t}_{c-1}, t_c) = \sum_{e=1}^{c+1} \sum_{t \in S_e} (\tilde{r}_t - b_0^{(e)} - b_1^{(e)} t)^2, \tag{3.20}$$

where $b_0^{(\cdot)}$ and $b_1^{(\cdot)}$ are the coefficients of the respective linear models. Once the definition of SSE is revised, the rest procedure for NPSD is adapted to select the significant change points in $\tilde{r}_t$. Figure 3.6(a) and Figure 3.6(b) present the intermediate detection results in our example while using the median size.

The key tuning parameter in this selection procedure is θ. In our application, we set $\theta = 0.5$ for both NPSD and the median particle size. The choice of $\theta = 0.5$ means that we deem a candidate a genuine change point if its selection reduces the SSE by half or more. By this choice, we detect one change point in NPSD and another one in median size; the two change points are shown as “#1" in Figure 3.5(c) and “#3" in Figure 3.6(c), respectively. For future applications, we would recommend the same choice for θ.

Setting $\theta = 0.5$, the change point detection method produces altogether two phase



(a) (b) (c)

Figure 3.6: Results of the proposed change point detection using median particle size: (a) 15 potential change points detected by 'strucchange' package; (b) change in $V(\cdot)$ when selecting a change point at a time; (c) all change points detected when θ is varied in the range of (0.2, 0.8).

change points: at 25.8s in $\tilde{r}_t$ ("#3" in Figure 3.6(c)) and at 39.9s in NPSD ("#1" in Figure 3.5(c)). These two change points segment the whole growth trajectory into three stages: (0s, 25.8s), (25.8s, 39.9s) and (39.9s, 76.6s). The delineated stages make it immediately clear how the nanocrystals grow: they go through two major growth stages with a transition stage in between. For this particular process, the two dominating mechanisms have been studied and understood [1, 75, 76, 96]: in the period of (0s, 25.8s), the orientated attachment (OA) mechanism dominates, whereas in the period of (39.9s, 76.6s), the Ostwald ripening (OR) mechanism dominates. It is understandable that the mechanism change does not happen suddenly. As one mechanism gradually takes over from the other, a short transition period naturally exists, which is the period of (25.8s, 39.9s) in this example.

3.5 Sensitivity of Tuning Parameter θ

Given the critical role played by θ, we further conduct a sensitivity analysis. Figure 3.7 shows the number of change points detected in both NPSD and median size, as θ varies in the range of $(0.2, 0.8)$. The NPSD-based detection produces either one change point or

two change points. The first change point detected in the two-point circumstance is the same as the change point detected in the single-point case, shown as “#1" in Figure 3.5(c). The second change point is shown as “#2" in the same figure. The median size based detection is more sensitive to the value of θ: it could produce from zero to two change points over the same θ range. The two change points that could have been detected are marked as “#3" and “#4", respectively, in Figure 3.6(c). Aside from the sensitivity issue, another drawback of using the median size statistic is that one would not be able to detect “#1" unless setting θ to some extreme value (like 0.1); given the analysis done by Zheng et al. [1], we know that a phase change indeed occurred around the time of “#1," so that missing this change point is a serious limitation.

When looking closely at the four possible change points, it is apparent to us that the change points “#2" and “#3" are the outcome of the same change, as their time stamps are only 3.2 seconds apart. By merging “#2" and “#3," the change point detection outcomes could possibly segment the whole growth into four stages, three stages, or two stages, depending the specific choice of θ. But an important message, we believe, is that the difference in the detection outcome does not lead to a drastically different understanding of the basic science behind. To see this point, consider the following alternatives.

When a smaller θ is used, all four change points could have been detected. Having “#4" apparently suggests the existence of an initial nucleation stage, which is generally hard to observe because its duration is short, data variability is high, and the number of nanocrystals is small. Missing this initial stage is understandable and not seriously detrimental to the subsequent analysis.

Had we chosen a large θ (say, greater than 0.6), only one change point (#1 in Figure 3.5(c)) would be detected in NPSD and no change point in median size. Hence, the transition stage could have been missed. Still, we would not miss the big picture of two dominating growths, i.e., OA and OR.

The overall analysis shows that NPSD-based detection outcome is robust, as it captures the important change points consistently in a broad range of the tuning parameter. To avoid missing potentially important change points in future applications, one should vary θ in a reasonable range and then chooses a manageable number of the change points.

The fact that NPSD-based detection produces a rather robust detection separating the whole growth trajectory into two major stages speaks to the benefit of having such a detection approach. Had we not known the individual mechanisms under respective stages, this detection outcome would hint strongly where to explore for understanding the basic science behind.

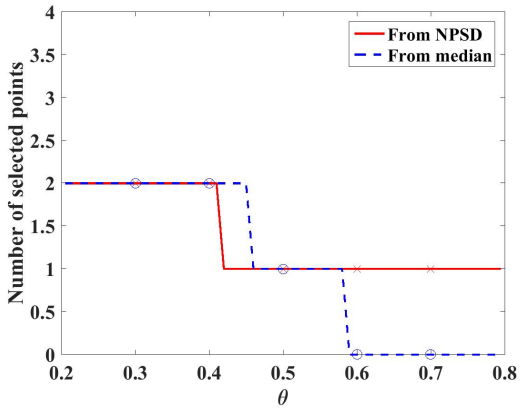


Figure 3.7: The number of change points detected in NPSD and median particle size for $\theta \in (0.2, 0.8)$.

3.6 Hybrid Modeling

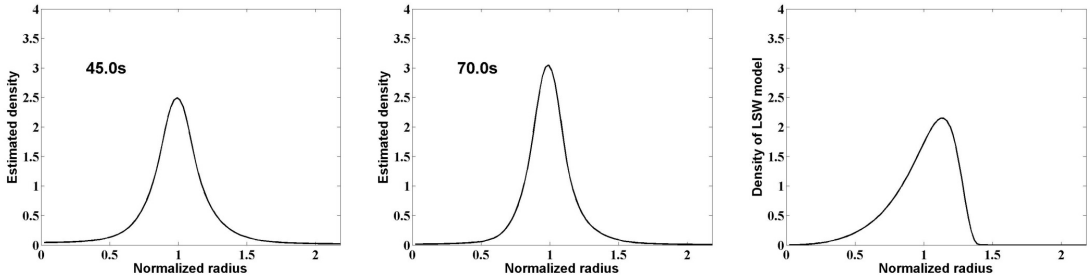
In this application, since we do know the dominating growth mechanisms, we can adopt the existing first principle models for each respective growth stage and then use an interpolation to model the transition period. As such, we produce a unified growth

model, as a hybrid of the first principle-based model and the empirical model, for the whole nanocrystal growth trajectory.

The models for NPSD $\hat{F}_t(\phi)$ and mean particle size $\bar{r}_t$ during the OA growth in the first stage (0s, 25.8s), taken from the work of Aldous [83], are, respectively:

$$\begin{aligned} F_{t_{\mathrm{OA}}}(\phi) &= \tfrac{2W_{\mathrm{OA}}}{\Gamma(a_{\mathrm{OA}}+1)}(W_{\mathrm{OA}}\phi)^{2a_{\mathrm{OA}}+1}e^{-(W_{\mathrm{OA}}\phi)^2}, \\ \bar{r}_{t_{\mathrm{OA}}}^{2(a_{\mathrm{OA}}+1)} &= b_{\mathrm{OA}}(t-t_{\mathrm{OA}}), \end{aligned} \tag{3.21}$$

where $W_{\mathrm{OA}} = (a_{\mathrm{OA}}+1)\Gamma(a_{\mathrm{OA}}+3/2)/\Gamma(a_{\mathrm{OA}}+1)$. Altogether three parameters used in the two models are: a_{OA}, indicating the variance of the process, b_{OA}, indicating the growth rate, and t_{OA}, indicating the initial size of nanocrystals.



(a) Estimated NPSD at 45s

(b) Estimated NPSD at 70s

(c) NPSD of the LSW model

Figure 3.8: (a) The empirical NPSD estimated at 45s; (b) the empirical NPSD estimated at 70s; (c) the theoretical NPSD derived from the LSW model.

The kinetics of OR growth in the third stage (39.9s, 76.6s) was usually described by the LSW model [82]. We did choose to use the LSW model to represent the mean particle size ($\bar{r}_t$) growth in the OR stage. For the $\bar{r}_t$ growth, the LSW model is to model the cube of $\bar{r}_t$ with a linear function. The model of $\bar{r}_t$ growth in the OA stage bears a similar appearance but the key difference is the different power term on $\bar{r}_t$.

However, to model $\hat{F}_t(\phi)$ in the OR growth part, we find that the LSW model cannot obtain a good fit for the estimated $\hat{F}_t(\phi)$. In Figure 3.8, we compare the empirical NPSDs estimated at 45s and 70s with the NPSD derived from the LSW model. The two empirical NPSDs are similar, and both of them look rather symmetric. The LSW-based NPSD is more skewed with a long lower tail and has larger variance compared with the empirical NPSDs. The long lower tail of the LSW-based NPSD presents a clear contrast with the NPSDs estimated directly from the data. In our opinion, there are two reasons for the mismatch. First, the smaller particles are difficult to track under the current resolution of the *in situ* TEM, yet the LSW model, with a long left tail, is more sensitive to the missed detection of these particles. Second, the LSW model has been known as inconsistent with the experimental results for a long time [97]. For this reason, other researchers proposed modified models to improve the fitting accuracy [98, 99, 84], but when these models are tested against the TEM video data at hand, they do not produce more competitive fitting quality. Here, we decide to use the OA growth model structure (derived by the Smoluchowski equation) to fit for NPSD in the OR growth; doing so indeed produces a better fit. The added benefit of using the same model structure in both stages is to make their comparison easier.

Specifically the OR growth models are:

$$\begin{aligned} F_{t_{\text{OR}}}(\phi) &= \tfrac{2W_{\text{OR}}}{\Gamma(a_{\text{OR}}+1)}(W_{\text{OR}}\phi)^{2a_{\text{OR}}+1}e^{-(W_{\text{OR}}\phi)^2}, \\ \bar{r}^3_{t_{\text{OR}}} &= b_{\text{OR}}(t - t_{\text{OR}}). \end{aligned} \tag{3.22}$$

The first equation here is the same as that in Equation (3.21) but with different parameters. The three parameters used in the OR models share the same interpretations as those in the OA model.

Using the TEM data, we estimate the parameters associated with the two stages, pre-

sented in Table 3.1. Compared with a_{OA}, the larger a_{OR} suggests that the larger variance of NPSD in the OR growth. This conclusion is consistent with the observations made by Zheng et al. [1], but our result provides a quantitative contrast. Using the estimated values of b_{OA} and b_{OR}, we calculate the derivative of $\bar{r}_t$ for the two stages. For the OA growth, the derivative is calculated as:

$$\frac{d\bar{r}_t}{dt} = \frac{1}{2(a_{\text{OA}}+1)} b_{\text{OA}} [b_{\text{OA}}(t - t_{\text{OA}})]^{\frac{1}{2(a_{\text{OA}}+1)}-1}, \tag{3.23}$$

and for the OR growth, the derivative is calculated as:

$$\frac{d\bar{r}_t}{dt} = \frac{1}{3} b_{\text{OR}} [b_{\text{OR}}(t - t_{\text{OR}})]^{\frac{1}{3}-1}. \tag{3.24}$$

Table 3.1: The estimated parameters associated with the two stages in the nanocrystal growth.

a_{OA}	b_{OA}	t_{OA}	a_{OR}	b_{OR}	t_{OR}
1.47	42.2	-429.3	7.31	0.55	-1342.7

In Figure 3.9, we compare the derivatives for the OA and OR growth. The gap between the two curves corresponds to the transition period in which no theoretical model is yet available. The two curves make it clear that in the nanocrystal growth, the mean radius growth rate in the OA stage is faster than that in the OR stage, just as the estimated b_{OA} and b_{OR} values suggested. This was again stated by Zheng et al. [1] but our analysis provides a quantitative picture of the mean radius evolution in the two stages.

The difference in t_{OA} and t_{OR} suggests that the initial nanocrystal sizes are different, and a more negative quantity implies a large initial size. The t_{OA} and t_{OR} values in Ta-

ble 3.1 make perfect sense, as the OR growth follows the OA growth, so that the initial nanocrystals in OR have a bigger size.

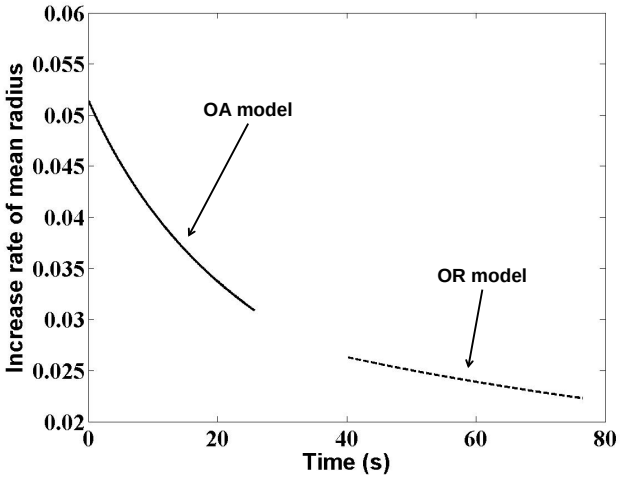


Figure 3.9: The comparison of the first derivative of $\bar{r}_t$ in the OA and OR growth stages.

To include the transition period between (25.8s, 39.9s), we introduce the weighting functions $\lambda_N(t)$ and $\lambda_R(t)$, for NPSD and mean particle size, respectively, to combine the two aforementioned models. The two weighting functions take the value of 0 when $t < 25.8$s, 1 when $t > 39.9$s, and increase from 0 to 1 quadratically in between, with their quadratic function coefficients fitted from the corresponding NPSD or mean particle size in the transition period. The overall growth models of $F_t(\phi)$ and $\bar{r}_t$, respectively, are in this hybrid structure as:

$$\begin{aligned} F_t(\phi) &= (1-\lambda_N(t))F_{t_{\text{OA}}}(\phi) + \lambda_N(t)F_{t_{\text{OR}}}(\phi), \\ \bar{r}_t &= (1-\lambda_R(t))\bar{r}_{t_{\text{OA}}} + \lambda_R(t)\bar{r}_{t_{\text{OR}}}. \end{aligned} \tag{3.25}$$

To verify the quality of our hybrid growth model, we show in Figure 3.10(a) the SSE

values between the $F_t(\phi)$ simulated using Equation (3.25) and its empirically estimated counterpart using directly the TEM observations. Except in the beginning few seconds and the transition period, the simulated results follow very closely the empirical results. The relatively worse fit during the transition period is understandable, as there lack theories to describe the transition mechanism. We also fitted Woehl et al. [2]'s single-stage model and show its SSE in Figure 3.10(a), too. Our hybrid model produces smaller SSE's for both the OA and OR growth stages and it is comparable to Woehl et al.'s model in the transition period.

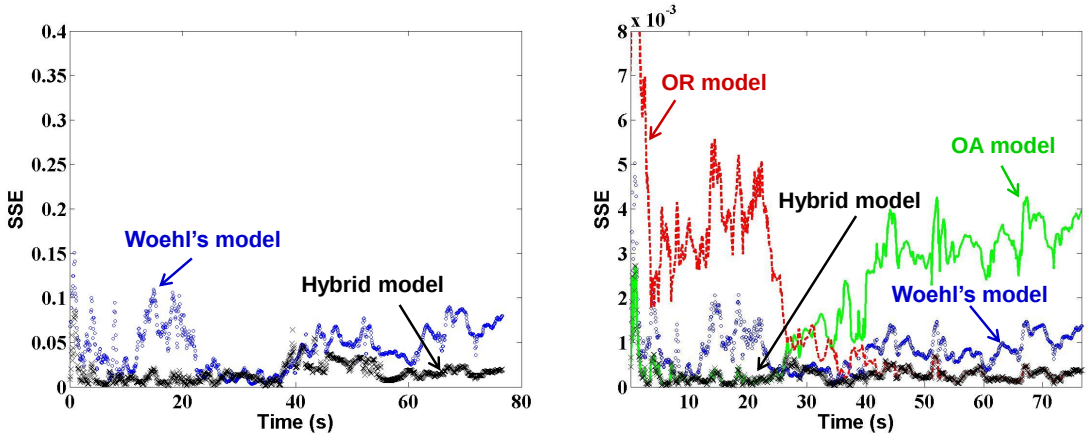
The above learning results provide a quantitative model to describe the whole growth trajectory. Using the learned results, we can simulate the evolution of PSD, $G_t(r)$, using the hybrid model of $\bar{r}_t$ and $F_t(\phi)$, as:

$$G_t(r) = \frac{1}{\bar{r}_t} F_t(\frac{r}{\bar{r}_t}). \tag{3.26}$$

Then, we estimated the PSD based directly on the observations of r_{st} by using the proposed non-parametric density estimation method. The SSE curve between the simulated PSD and the estimated PSD is shown in Figure 3.10(b). Additionally, we also show the SSE curves between the estimated PSD and the PSDs simulated by using, respectively, Woehl et al. [2]'s single-stage model, the OA growth model alone and the OR growth model alone. The hybrid growth model fits the observed data consistently well throughout the entire growth trajectory, while other models all have deficiencies in certain periods.

3.7 Concluding Remarks

In this chapter, we propose a method aiming at identifying and delineating different stages in a nanocrystal growth using *in situ* TEM videos, assuming the self-similar analytic solution existed for a fixed growth mechanism. We make two major contributions: the first is to estimate a time-varying NPSD by pooling data of all time frames and develop a



(a) SSE of the NPSD models

(b) SSE of the PSD models

Figure 3.10: The comparison of the simulated results and the empirical estimation from the data: (a) the SSE curves between the simulated NPSD (by using the hybrid model or Woehl's model) and its estimated counterpart; (b) the SSE curves between the simulated PSDs (by using the hybrid model, Whoehl's model, OA model alone, and OR model alone, respectively) and their estimated counterpart.

modified penalized B-spline method accordingly; the second is to perform a robust change point detection of the highly stochastic nanocrystal growth process, providing a detection outcome consistent with physical understanding. We applied our change point analysis to a published *in situ* TEM video clip.

Our work shows that the importance of using probability distribution functions, not the simple statistics, for the phase identifying and model building purposes. It also reveals the existence of a transition period between the two dominating growth stages. The existence of the transition period is expected and our method finds its precise timing. But the underlying mechanism of the transition period was still poorly understood. Our hybrid model, assuming a linear combination of the two dominating growth mechanisms in the transition period, provides an initial attempt and fits the observations reasonably well. Moreover, the estimated time-varying NPSD gives another evidence that the LSW model does not fit experiment results well at the latter stage. We hope that our method can help the material

scientists find an accurate theoretical model for predicting the long-time distribution of the Oswald ripening. Overall, we believe that our detection and modeling efforts lay a foundation for future quality control of nanocrystal synthesis processes. With the density estimation and the predictive model, engineers can monitor the process and detect the out-of-controls by comparing the observed and theoretical distributions.

4. FAST DYNAMIC NONPARAMETRIC DISTRIBUTION TRACKING IN ELECTRON MICROSCOPIC DATA

In this chapter, we will extend the retrospective analysis in Chapter 3 to a prospective analysis. To enable in-process control of nanocrystal production, taking full advantage of *in situ* TEM hinges upon a solution addressing a statistical challenge, which is the capability of tracking a dynamic, time-varying probability distribution reflecting the nanocrystal growth. Because no known parametric density functions can adequately describe the evolving distribution, a nonparametric approach is inevitable. Towards this objective, we propose to incorporate the dynamic evolution of the normalized particle size distribution into a state-space model, in which the distribution curve is represented by a B-splines method. A closed-form algorithm runs online updates faster than the frame rate of the *in situ* TEM video, making it suitable for in-process control purposes. By imposing the constraints of the curve smoothness and temporal continuity, we improve the accuracy and robustness of the estimation. We test our method on three published TEM videos. For all of them, our proposed method is able to obtain efficient and accurate estimations and outperforms several alternative approaches.

4.1 Introduction

A promising method of producing nanocrystals in large quantities is a self-assembly process, referring to the process of producing nanocrystals from small building blocks such as atoms and molecules that are spontaneously arranged into order structures at the nanoscale [3, 4]. To produce nanocrystals with desired sizes and shapes, its growth process should be monitored and controlled [100], but accomplishing this goal is rather challenging, due to the existence of multiple growth mechanisms [1], complex interactions among hundreds of nanoscale particles [16], and after all, the stochastic nature of

the growth processes. Critical to the mission of achieving in-process control is a recent technology innovation in nanoscale metrology, the *in situ* TEM [1]. An *in situ* TEM uses a special sample holder in which a nanocrystal growth process takes place, allowing motion pictures to be taken while the nanocrystals in the sample holder are initiating, crystalizing, and morphing into different sizes and shapes.

The morphological features to be extracted from a TEM video are the sizes and shapes of the nanocrystals and their evolving trajectories over the time. In this study, we focus primarily on particle size, because all the TEM videos we have at hand contain nanocrystals of rather uniform round shape throughout their growth process. We note that the current progress by research communities in handling *dynamic* TEM images (i.e., videos) is still at the stage of dealing with size, rather than both size and shape.

When an image frame of the nanocrystal growth process is recorded by an *in situ* TEM, an image processing tool is used to extract the contours of the nanocrystals in the frame, count the quantity, and calculate the particle sizes. After that, a histogram of the normalized particle size distribution (NPSD) is created and used as the observational input to the subsequent modeling. Here, the NPSD is the original particle size distribution normalized by the average radius of the nanocrystals at a given moment. It is understandable that the trajectory of the particle size distribution is trending upward, as the nanocrystals are getting bigger over the growth process. Once normalized by the average particle size, the upward trend is eliminated in the normalized particle size distribution, making it easier to be monitored.

Studies show that tracking the time-varying NPSD can indeed provide valuable insights to unearth the underlying growth dynamics and mechanisms [1, 2]. As such, the nano production monitoring problem is translated to a statistical learning problem, which is to model and track a time-varying probability density function (PDF), subject to the following requirements.

The first requirement is rooted in that no known parametric density functions can adequately describe the time-varying NPSD throughout the nanocrystal growth. A nonparametric approach appears inevitable. The second requirement is the need for a prospective analysis, because our goal is online monitoring and tracking, and only through this online capability does it enable in-process control. For an online analysis, the model updating to capture the PDF change needs to be fast enough; how fast is enough is dictated by the imaging speed (in this application, about 15 frames per second). The last requirement arises from that the time-varying NPSD needs to be a smooth function at each frame while also maintain smoothness and continuity across frames. Imposing these two constraints, referred to as curve smoothness and temporal continuity respectively, will improve the robustness and accuracy of the estimation, especially when the number of observations is not large enough at individual image frames.

To address these technical challenges, we characterize the dynamics of the normalized particle size distribution with a state-space model, in which each PDF is representably nonparametrically by B-splines. To enable online analysis, a closed-form estimation approach is devised to update the PDF when new observations comes, so that tracking can run fast enough to catch up with the imaging rate. While the temporal continuity is naturally imposed in the proposed model, imposing the curve smoothness in the dynamic model is not trivial. We are able to accomplish the objective by introducing a new state vector that penalizes the second order difference of the B-splines coefficients.

The remaining parts of the chapter are organized as follows. In Section 4.2, we review the related work. Section 4.3, we discuss the data used in this study. In Section 4.4, we present the state-space model and devise a closed-form for online updating and tracking of the particle size distribution. In Section 4.5, we explain how to estimate the parameters used in the state-space model. In Section 4.6, we apply our method to the analysis of three segments of TEM videos and demonstrates the merits of the proposed method. Finally, we

conclude our work in Section 4.7.

4.2 Related Work

Two branches of research are related to our undertaking: (1) nano image processing, (2) online estimation of nonparametric density functions. We review both in the following.

Nano imaging processing. The vast majority of the existing methods for analyzing TEM measurements, including several of our own, are for handling still images [7, 9, 37, 88]. These methods laid the foundation for handling dynamic images in TEM videos. One can even use them to process the images one frame at a time. Of course, processing one frame at a time is inefficient and also overlooks the dynamics and correlation among the adjacent video frames.

There are a few approaches available for handling dynamic TEM images, and they fall into two major lines of approach. The first line is to identify and track individual nanocrystals [81, 16] and build a model to characterize the growth dynamics by looking at those trajectories. This approach is in line with the objects/targets tracking research in the computer vision literature [13, 14]. The second line is to model the distribution of certain characteristics of the objects (say, size), instead of tracking individual objects.

The second line of approach, i.e., the distribution tracking, is more common in material science, which is in fact what we follow in this research. On the one hand, material scientists care more about the collective changes in the distribution of nanocrystals, because the distribution change has a good scientific ground to be connected with the underlying process dynamics, while the change exhibiting in any single nanocrystal may not be representative. On the other hand, it is rather difficult, if not impossible, to track individual nanocrystals in TEM videos and link them across the image frames. Individual nanocrystals lack traceable features. In certain *in situ* TEM systems, samples continuously flow through the imaging area, so that the samples observed every time are different, for which

individual object tracking is no longer meaningful.

Online estimation of a nonparametric distribution. Many existing methods, e.g., the smoothed histogram [101], the Kernel estimator [17], and penalized B-splines [18], can fit a smooth curve for a static distribution. However, employing those methods to estimate the distribution at each time individually fails to capture the dynamics between video frames. Modeling a time-varying distribution is an open and challenging statistical learning problem.

Some recent research developments, e.g., [19, 20, 21, 22], outline a dynamic hierarchical framework to address the time-varying PDF modeling problem. Under this framework, a latent variable, changing through time, underpins the dynamics of the underlying process and drives the distribution to change over time. In the engineering fields, such a framework is usually called a state-space model, where the latent variable is regarded as a time-varying hidden state. To implement such a framework, the following two questions ought to be answered: one is what to be defined as the hidden state and how to connect it with the time-varying distribution, and the second is how to efficiently update the state, and thus the distribution, upon receiving new observations, for the purpose of online tracking?

To our best knowledge, the existing works generally follow two major schools of thought. One school of thought [19, 21] is to directly use the histogram as an approximation of the true distribution. In doing so, one can segment the range of a variable of interest into several intervals, and calculate the number of observations within each interval to build a histogram for each frame of image. The means of the intervals are regarded as the state, assumed to follow an autoregressive model. After the new observations come, the state will be updated through a particle filtering process [23]. The drawbacks of this approach are its sensitivity to the number and positions of the intervals, and the high computational complexity when the dimension of the state is large.

The other school of thought [20, 22], assumes that the prior of the time-varying distribution follows a Dirichlet mixture model [102]. This assumption treats the distribution as the summation of an infinite number of local kernels, where the weights and locations of those kernels are drawn from a dependent, time-varying Dirichlet process. The parameters of the Dirichlet process can be considered as the state, and change over time through a random walk or a diffusion process. The inference for the state is implemented by a posterior sampling according to the new observations. Such an approach also suffers the high computational complexity in the posterior sampling, especially when a large number of local kernels exist in the model. Generally speaking, the sampling based approaches are too slow to meet the online tracking objective (recall the 15 frames per second imaging rate).

To develop a distribution estimation method suitable for online video tracking, the state at each time should be updated by a fast, closed-form algorithm. In that way, the computational cost will not increase dramatically with the dimension of the states. In a previous *retrospective analysis* of TEM videos [103], it was shown that the B-splines model [18] provides a competitive method for estimating a nonparametric, time-varying PDF. The PDF curve is represented by a linear combination of a set of B-spline basis functions, the coefficients of these basis functions, regarded as the hidden states, are efficiently estimated when the density curve is fit to the observed histograms. To extend the previous retrospective approach for the online mission, we model the change of the coefficients with a random walk, and build a state-space for the time-varying distributions. A closed-form state updating is then accomplished by devising an extended Kalman filter [104] based on the proposed prospective model.

To alleviate the sensitivity to the number and positions of the intervals in the observed histograms, both the curve smoothness and temporal continuity need to be imposed on the proposed model. While the random walk of the hidden state naturally provides the

temporal continuity, a technical challenge to be addressed is how to incorporate the curve smoothness in the dynamic model. One may just want to add the smoothness constraint onto the hidden state, say, imposing a constraint on the correlation of the state vector. But doing so makes both the updating and parameter estimation tasks difficult to carry out. Our proposed solution is to follow a suggestion made by Eilers and Marx [18], which is that one can penalize the second-order difference of the B-spline coefficients for imposing smoothness on a resulting curve. Specifically, we propose to introduce a new state vector and make part of the new vector equivalent to the second-order difference of the original state vector. Then, we require the state variables corresponding to the second-order difference to be much smaller than the other part of the new state vector; such action provides us a much smoother and better distribution estimation and its inclusion does not slow down the otherwise fast state updating via the extended Kalman filter.

Comparing with the alternatives for the time-varying distribution estimation, our proposed method provides a fast and closed-form updating algorithm for the hidden state meeting the online video tracking requirement, while imposing the two constraints on the B-splines leads to a robust and accurate estimation. Figure 4.1 presents an overview of the proposed online, prospective analysis.

4.3 Data

A newly emerged technology and rather expensive, there are not many *in situ* TEMs available yet in the United States. There are a very limited number of TEM videos available in the public domain. In this study, we use three clips of *in situ* TEM video: two clips published by Zheng et al. [1] and one clip published by Woehl et al. [2]. The three video clips have, respectively, 1,149, 637, and 112 image frames. We label them as Video 1, Video 2 and Video 3, respectively. Figure 4.2 presents four frames of Video 1, capturing the growth of platinum nanocrystals.

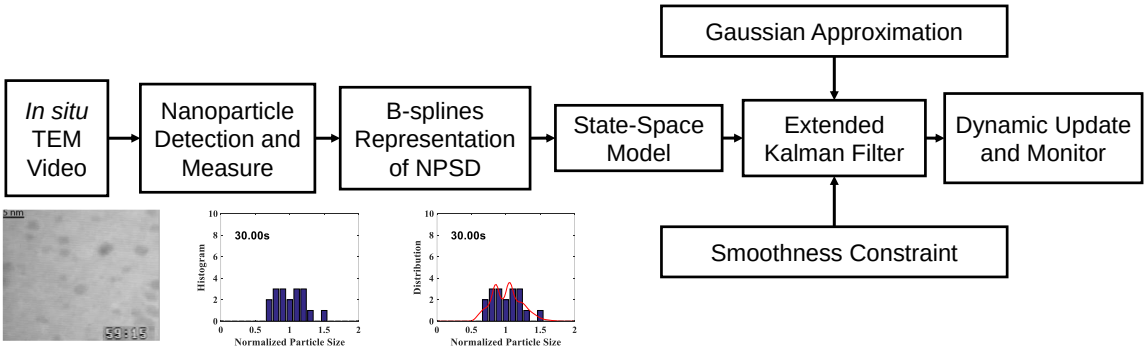


Figure 4.1: The framework of a prospective analysis of *in situ* TEM videos.

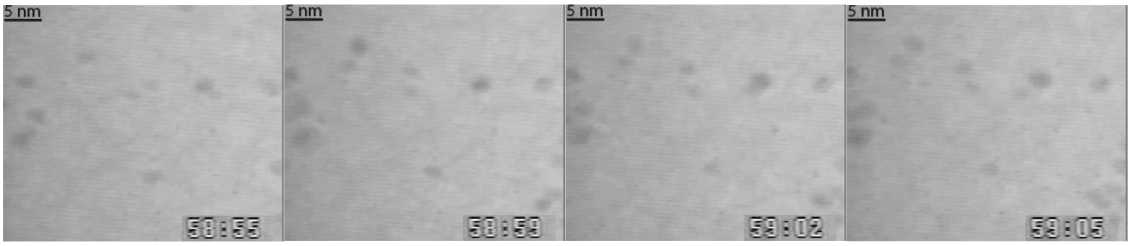


Figure 4.2: Four frames from the *in situ* TEM video studied by [1]. The dark spots are nanocrystals.

The data processing works as follows. When an image frame of the nanocrystal growth process is recorded by an *in situ* TEM, we first process the image and extract the nanocrystal information, which is the number and the corresponding size of the nanocrystals in the frame. The specific tool for processing individual images is from [88], a method particularly potent of handling noisy TEM images with low contrast. The result of one frame from each video clip is shown in Figure 4.3. Of the three video clips, Video 1 and 2 are of 290×242 pixels in size and Video 3 is of 496×472 pixels. Considering their relatively small image size, the image pre-processing can be done fairly quickly. For Video 1 and Video 2, the image processing only takes 0.04 seconds per frame and for Video 3 it takes 0.02 seconds per frame.

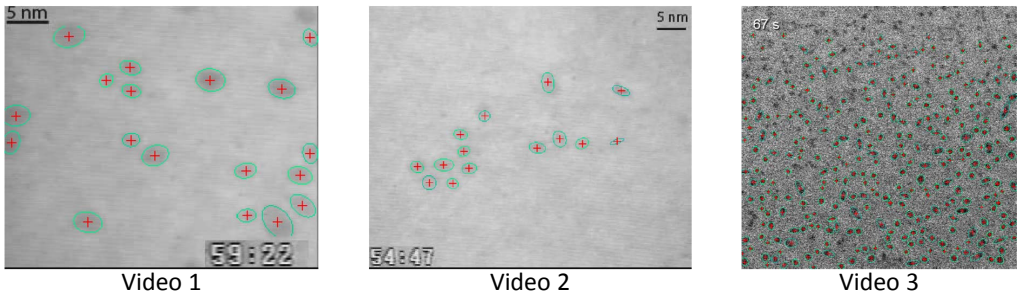


Figure 4.3: The nanocrystal detection results from a single frame from the three clips of our tested video, where the green line shows a nanocrystal's edge and the red '+' shows a nanocrystal's center. Videos 1 and 2 were published by [1] and Video 3 was published by [2].

After we detect all nanocrystals in the frame of time t, we calculate each nanoscrystal's area, $A^{\ell}(t)$, for the ℓ-th nanocrystal at time t, for $\ell = 1, ..., N_t$, where N_t is the total number of nanoscrystals in the frame of time t. Then, we use $A^{\ell}(t)$ to compute the average radius of the ℓ-th nanocrystal, namely $r^{\ell}(t) = \sqrt{A^{\ell}(t)/\pi}$. The mean radius for each image frame, $\bar{r}(t)$, can be readily obtained. Finally, we normalize $r^{\ell}(t)$ by $\bar{r}(t)$ to obtain the normalized

radius $x^{\ell}(t)$, such that $x^{\ell}(t) = r^{\ell}(t)/\bar{r}(t)$, which are further used to create a histogram.

To create the histogram, we limited the range of $x^{\ell}(t)$ to $[0, 2.0]$, as the number of the nanocrystals twice as large as the average size is usually very few at any given time. We divide the range into m intervals of equal size δ. Here we use a constant $m = 21$ throughout the monitoring process and denote by x_i the normalized particle size corresponding to the center of the ith interval, $i = 1, \ldots, m$. The resulting histogram for the frame of time t is denoted by the vector of $\mathbf{Y}_t = [Y_{1t}; Y_{2t}; \cdots; Y_{mt}]$, where Y_{it} is the number of the observed $x^{\ell}(t)$'s falling into the ith interval of the histogram. In Section 4.6.2, we will investigate the sensitivity of the proposed method to the parameters of the input histogram.

4.4 State-Space Modeling and Updating

4.4.1 State-Space Model for Normalized Particle Size Distribution

Our primary objective is to estimate the normalized particle size distribution function $f_t(x)$ of $x^{\ell}(t)$. For this purpose and at a given time t, we follow the procedure in [18], which derives a smooth representation of $f_t(x)$ fitting the histogram data $\mathbf{Y}_t$. This modeling choice is made mainly because doing so allows us to impose both the curve smoothness and temporal continuity in an efficient way on the time-varying distribution; more details will be discussed in the subsequent sections.

As Y_{it} is the count of observations falling in $x_i - \delta/2 < x^{\ell}(t) < x_i + \delta/2$, it is a standard approach to assume that Y_{it} follows the Poisson distribution with expectation λ_{it} [105]:

$$Y_{it} \sim \text{Poisson}\{\lambda_{it}\},\ i = 1, \ldots, m. \tag{4.1}$$

Following the treatment used by Eilers and Marx [18], we model the count data with B-splines. We adopt a generalized linear model with a log link function to represent λ_{it}

as:

$$\log \lambda_{it} = \sum_{j=1}^{n} \alpha_{jt} B_j(x_i), \tag{4.2}$$

where n is the number of basis functions, $B_j(x)$ is the B-spline jth basis, and α_{jt} is its coefficient at time t. The log link function can guarantee a positive λ_{it}. Collectively, $[\alpha_{1t}; \alpha_{2t}; \cdots; \alpha_{nt}]$ can be represented as a vector $\boldsymbol{\alpha}_t$. If we write the B-spline basis functions as a matrix $\mathbf{B}$, such that $(\mathbf{B})_{ij} = B_j(x_i)$, the Poisson model of Y_{it} will be expressed as:

$$Y_{it} \sim \text{Poisson}\{(\exp[\mathbf{B}\boldsymbol{\alpha}_t])_i\},\ i = 1, \ldots, m. \tag{4.3}$$

Once $\boldsymbol{\alpha}_t$ is estimated, we can further obtain the normalized particle size distribution $f_t(x)$ as follows. As $\lambda_{it} \propto \int_{x_i-\delta/2}^{x_i+\delta/2} f_t(x)dx$, $\lambda_{it} \propto f_t(x_i)$ when δ is small, so that $f_t(x)$ can be estimated by the continuous form of Equation (4.2) as

$$f_t(x) = \frac{1}{\text{const}_t} \exp[\sum_{j=1}^{n} \alpha_{jt} B_j(x)], \tag{4.4}$$

where const_t is a constant to guarantee $f_t(x)$ integrating to one.

There is a rich literature on modeling counts data like $\mathbf{Y}_t$; for example, dynamic Poisson models [21, 106, 107] or multivariate Poisson regression [19, 108, 109]. Most of the methods update the states by posterior sampling, which is typically slow and hence cannot catch up with the frame rate of TEM video. They fall under the retrospective analysis category. We here look for a method that can be easily incorporated into the dynamic state-space model for a prospective analysis.

With the B-spline representation, we can simply use the B-spline coefficient vector, $\boldsymbol{\alpha}_t$, as the state vector in the proposed state-space model, because the change in $\boldsymbol{\alpha}_t$ indicates the change of the underlying normalized particle size distribution $f_t(x)$. Previous studies [82, 83] show that $f_t(x)$ undergoes small fluctuations during a growth stage in which the

commanding physical growth mechanism remains the same, while $f_t(x)$ will see a much greater change as a different growth mechanism takes over. Based on this understanding, we assume that the state vector, $\boldsymbol{\alpha}_t$, follows a random walk. The step size of the random walk is small under a given growth mechanism, but the step size gets large during the period as the nanocrystal growth transitions from one mechanism to another, meaning that the random walk takes a great stride to catch up with the changes in the underlying process. In Figure 4.7 presented later about the innovation sequence, this change pattern in the step size of the random walk is confirmed.

As such, the state equation reads as:

$$\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1} + \mathbf{w}_t, \tag{4.5}$$

where $\mathbf{w}_t$ is the disturbance vector of the state and assumed to follow the distribution of normal$(\mathbf{0}, \mathbf{Q})$. The covariance matrix $\mathbf{Q}$ will be treated as a constant matrix throughout the process. The state updating equation (4.5) and the observation equation (4.3) constitute our state-space model.

4.4.2 Online Updating of State $\boldsymbol{\alpha}_t$

Updating the estimation of NPSD online is thus equivalent to updating the state vector in the state-space model. In the dynamic systems and control theory, the Kalman filter [110] is arguably the most popular method used for conducting such update. For linear state-space models with Gaussian observations, a Kalman filter [110] uses the posterior mean $\mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{Y}_1, \cdots, \cdots, \mathbf{Y}_t)$, denoted as $\hat{\boldsymbol{\alpha}}_t$, to iteratively estimate $\boldsymbol{\alpha}_t$. There are two main steps in a Kalman filter. The first step, known as the prediction, is to predict the prior estimator $\hat{\boldsymbol{\alpha}}_t^-$ and covariance matrix $\mathbf{P}_t^-$ of the state at time t, based only on the observations received up to time $t-1$. When the new observation of $\mathbf{Y}_t$ arrives, the Kalman filter undertakes a correction step to obtain the posterior estimator $\hat{\boldsymbol{\alpha}}_t$ and covariance matrix $\mathbf{P}_t$.

For a Gaussian system, the Kalman filter has a closed-form solution for both the prediction and correction steps and can thus run very efficiently.

Unfortunately, our state-space model of the time-varying NPSD is not a Gaussian system because $\mathbf{Y}_t$ follows a Poisson distribution in (4.3). To solve for the posterior mean $\mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{Y}_1, \cdots, \mathbf{Y}_t)$, one possible solution approach is to use the sampling methods, such as particle filtering [23], to simulate the posterior distribution of the state $\boldsymbol{\alpha}_t$. But the sampling approach is not suitable for the online estimation objective because the approach's computational speed can hardly meet the online update requirement. After knowing the model set up, the shortcoming of the sampling approach is even more obvious. To estimate the NPSD accurately, both $\mathbf{Y}_t$ and $\boldsymbol{\alpha}_t$ should have a moderate to high dimension; for instance, $m \geq 10$ and $n \geq 10$. To sample from a space of such dimension for approximating a posterior distribution, the sample size are rather large, making its computational efficiency a daunting task to be addressed.

Our solution is to extend the Kalman filter by adopting Durbin and Koopman [111]'s method to find a good Gaussian approximation of the Poisson observation in (4.3) locally around the current estimation $\hat{\boldsymbol{\alpha}}_t$. Rather to approximate the Poisson globally, the Gaussian distribution will have a similar shape within the neighborhood of $\hat{\boldsymbol{\alpha}}_t$. As the Kalman filter usually updates $\boldsymbol{\alpha}_t$ near its current position, such a local approximation can lead to an efficient and accurate estimation. When used in our context, this means that we want to have the following approximation:

$$\mathbf{Y}_t \sim \text{normal}(\mathbf{B}\boldsymbol{\alpha}_t + \boldsymbol{\mu}_t, \mathbf{H}_t), \tag{4.6}$$

so that the probability density functions of Equation (4.3) and (4.6) have the same first and second derivatives with respect to $\boldsymbol{\alpha}_t$ near $\hat{\boldsymbol{\alpha}}_t$. Following this thought, we can derive the following expressions for the mean vector $\boldsymbol{\mu}_t$ and the covariance matrix $\mathbf{H}_t$ (please see the

derivation details in Appendix A.2):

$$\begin{aligned} \boldsymbol{\mu}_t &= \mathbf{Y}_t - \mathbf{B}\hat{\boldsymbol{\alpha}}_t - \exp(-\mathbf{B}\hat{\boldsymbol{\alpha}}_t)[\mathbf{Y}_t - \exp(\mathbf{B}\hat{\boldsymbol{\alpha}}_t)], \\ \mathbf{H}_t &= \text{diag}[\exp(-\mathbf{B}\hat{\boldsymbol{\alpha}}_t)]. \end{aligned} \tag{4.7}$$

As such, the original state-space model is converted into an approximated Gaussian state-space model, now constituting of Equation (4.6) and Equation (4.5). Technically, a standard Kalman filter can then be devised and applied.

A remaining problem is that $\hat{\boldsymbol{\alpha}}_t$ is unknown when we calculate $\boldsymbol{\mu}_t$ and $\mathbf{H}_t$ by Equation (4.7). We use an iterative strategy to find $\hat{\boldsymbol{\alpha}}_t$: first we use the prior estimator $\hat{\boldsymbol{\alpha}}_t^-$ to calculate $\boldsymbol{\mu}_t$ and $\mathbf{H}_t$, then update $\hat{\boldsymbol{\alpha}}_t$ by the Kalman filter; and after that, we update $\boldsymbol{\mu}_t$ and $\mathbf{H}_t$ using the newly estimated $\hat{\boldsymbol{\alpha}}_t$. Repeat this process until $\hat{\boldsymbol{\alpha}}_t$ converges. According to both [111] and our experiments, this process converges in a number of steps.

Algorithm 2 presents the detailed estimation and updating process. We put in Appendix A.3 the basic steps and explanations of the Kalman filter for readers who are not familiar with Kalman filter.

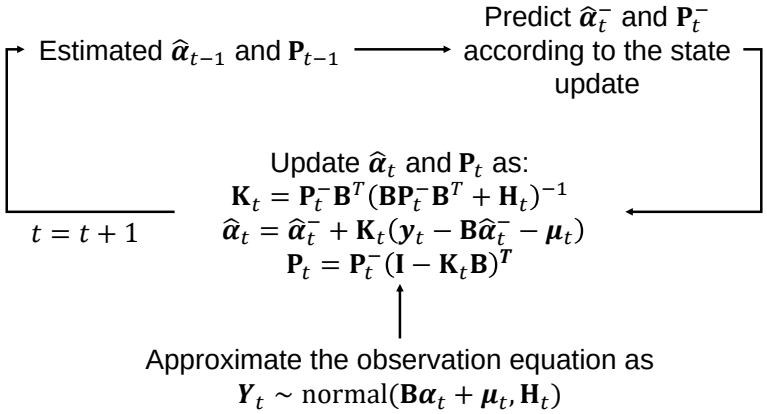


Figure 4.4: The illustration of main steps of online updating of the state $\boldsymbol{\alpha}_t$.

Algorithm 2 online updating method of the state-space model.

1. Set $t = 1$ and initialize $\hat{\boldsymbol{\alpha}}_0$ and $\mathbf{P}_0$.
2. Predict the prior estimator of the state as: $\hat{\boldsymbol{\alpha}}_t^- = \hat{\boldsymbol{\alpha}}_{t-1}$.
3. Predict the prior covariance matrix as: $\mathbf{P}_t^- = \mathbf{P}_{t-1} + \mathbf{Q}$.
4. Set $\hat{\boldsymbol{\alpha}}_t = \hat{\boldsymbol{\alpha}}_t^-$.
5. Calculate $\boldsymbol{\mu}_t$ and $\mathbf{H}_t$ as:

$$\boldsymbol{\mu}_t = \mathbf{Y}_t - \mathbf{B}\hat{\boldsymbol{\alpha}}_t - \exp(-\mathbf{B}\hat{\boldsymbol{\alpha}}_t)[\mathbf{Y}_t - \exp(\mathbf{B}\hat{\boldsymbol{\alpha}}_t)],$$
$$\mathbf{H}_t = \text{diag}(\exp(-\mathbf{B}\hat{\boldsymbol{\alpha}}_t)).$$

6. Computer the innovation and its covariance matrix:

$$\boldsymbol{\nu}_t = \mathbf{Y}_t - \mathbf{B}\hat{\boldsymbol{\alpha}}_t^- - \boldsymbol{\mu}_t; \;\; \mathbf{F}_t = \mathbf{B}\mathbf{P}_t^-\mathbf{B}^T + \mathbf{H}_t.$$

7. Computer the Kalman gain as: $\mathbf{K}_t = \mathbf{P}_t^-\mathbf{B}^T\mathbf{F}_t^{-1}$.
8. Update the posterior estimator with measurement $\mathbf{Y}_t$: $\hat{\boldsymbol{\alpha}}_t = \hat{\boldsymbol{\alpha}}_t^- + \mathbf{K}_t\boldsymbol{\nu}_t$.
9. Repeat Step 5 to 8 until $\hat{\alpha}_t$ converges.
10. Update the posterior covariance matrix as: $\mathbf{P}_t = \mathbf{P}_t^-(\mathbf{I} - \mathbf{K}_t\mathbf{B})^T$.
11. Set $t = t + 1$, repeat from 2 until $t = T$.

After we obtain the posterior estimation of the state $\hat{\boldsymbol{\alpha}}_t$, the corresponding NPSD $\hat{f}_t(x)$ can be represented as:

$$\hat{f}_t(x) = \exp[\sum_{j=1}^{n} \hat{\alpha}_{jt} B_j(x)], \tag{4.8}$$

where $\hat{\alpha}_{jt}$ is the jth element of $\hat{\boldsymbol{\alpha}}_t$. Figure 4.4 highlights the main steps of online updating of the time-varying NPSD.

4.4.3 Curve Smoothness for Distribution Estimation

While Algorithm 2 can provide an online estimation and updating of the time-varying NPSD, it does not impose any requirement on the smoothness of the estimated distribution. Without a proper smoothness constraint, our density estimation so far is sensitive

to choices like the number of intervals in the histogram $\mathbf{Y}_t$ and the number of B-spline basis functions, and could become considerably inaccurate in the cases that some middle intervals in the input histogram turn out empty. So our goal here is to incorporate the curve smoothness constraint and make it to work with the state-space model.

We plan to impose the curve smoothness constraint for the B-splines density estimation by penalizing the second order difference of the coefficients vector $\boldsymbol{\alpha}_t$, which is denoted as an $n-2$ dimension vector $\Delta^2\boldsymbol{\alpha}_t$ and defined as

$$\Delta^2\boldsymbol{\alpha}_t = \begin{bmatrix} -\alpha_{1t} + 2\alpha_{2t} - \alpha_{3t} \\ -\alpha_{2t} + 2\alpha_{3t} - \alpha_{4t} \\ \cdots \\ -\alpha_{(n-2)t} + 2\alpha_{(n-1)t} - \alpha_{nt} \end{bmatrix} = \begin{bmatrix} -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 2 & -1 \end{bmatrix} \boldsymbol{\alpha}_t. \tag{4.9}$$

When the magnitude of the difference, $||\Delta^2\boldsymbol{\alpha}_t||_2$, is small, it implies certain degree of smoothness in the resulting density curve $f_t(x)$.

To put a constraint on $||\Delta^2\boldsymbol{\alpha}_t||_2$, we propose to transform linearly the original state $\boldsymbol{\alpha}_t$ into another state $\boldsymbol{\gamma}_t$. The new state $\boldsymbol{\gamma}_t$ will conclude $\Delta^2\boldsymbol{\alpha}_t$ but have the same dimension as $\boldsymbol{\alpha}_t$. We make the last $n-2$ coordinates of $\boldsymbol{\gamma}_t$ equal to $\Delta^2\boldsymbol{\alpha}_t$, then add first two coordinates to make it a n dimension vector. A straightforward way of setting them is to let γ_{1t} as the summation of all the even coordinates of $\boldsymbol{\alpha}_t$, while γ_{2t} as the summation of all the odd coordinates of $\boldsymbol{\alpha}_t$. So the transformation from $\boldsymbol{\alpha}_t$ to $\boldsymbol{\gamma}_t$ can be written as:

$$\gamma_{1t} = \sum_{j=1}^{[n/2]} \alpha_{(2j)t},\ \gamma_{2t} = \sum_{j=1}^{[n/2]} \alpha_{(2j-1)t},\ \boldsymbol{\gamma}_{(3:n)t} = \Delta^2\boldsymbol{\alpha}_t, \tag{4.10}$$

where $[n/2]$ is the largest integer smaller than or equal to $n/2$. As such, we can write this

linear transform in a matrix format, such that $\boldsymbol{\alpha}_t = \mathbf{C}\boldsymbol{\gamma}_t$, where

$$\begin{aligned} \mathbf{C}^{-1}_{1(2j)} = 1,\ \mathbf{C}^{-1}_{2(2j-1)} = 1,\ j = 1, \cdots, [n/2]; \\ \mathbf{C}^{-1}_{j(j-2)} = -1, \mathbf{C}^{-1}_{j(j-1)} = 2, \mathbf{C}^{-1}_{jj} = -1, j = 2, \cdots, n; \end{aligned} \tag{4.11}$$

and other elements of $\mathbf{C}^{-1}$ are equal to 0.

By using the new state $\boldsymbol{\gamma}_t$, we can change the state-space model to:

$$\begin{aligned} Y_{it} &\sim \text{Poisson}\{(\exp[\mathbf{BC}\boldsymbol{\gamma}_t])_i\}, \\ \boldsymbol{\gamma}_t &= \boldsymbol{\gamma}_{t-1} + \mathbf{w}_t, \end{aligned} \tag{4.12}$$

where $\mathbf{w}_t \sim \text{normal}(\mathbf{0}, \mathbf{Q})$. Here we slightly abuse the notations – even though $\mathbf{w}_t$ and $\mathbf{Q}$ are used again, they are of different values from those in Equation (4.5).

This transformation in the state vector allows us to use the structure of $\mathbf{Q}$ to add the smoothness constraint on the estimated density $\hat{f}_t(x)$. As [18] pointed out, a small $||\Delta^2\boldsymbol{\alpha}_t||_2$ can give us a smooth $\hat{f}_t(x)$. Since the first two coordinates $[\gamma_{1t}, \gamma_{2t}]$ are the summation of the elements in $\boldsymbol{\alpha}_t$, and the remaining coordinates $[\gamma_{3t}, \cdots, \gamma_{nt}]$ are $\Delta^2\boldsymbol{\alpha}_t$, we hope that $[\gamma_{3t}, \cdots, \gamma_{nt}]$ should have smaller magnitudes than $[\gamma_{1t}, \gamma_{2t}]$. That means the corresponding disturbing vector $\mathbf{w}_t$ has the same property, i.e., $[w_{3t}, \cdots, w_{nt}]$ are much smaller than $[w_{1t}, w_{2t}]$. Given $\mathbf{w}_t \sim \text{normal}(\mathbf{0}, \mathbf{Q})$, we can conclude that in the covariance matrix $\mathbf{Q}$, the values related to $[w_{3t}, \cdots, w_{nt}]$ should be also smaller than those related to $[w_{1t}, w_{2t}]$.

Being aware that $[\gamma_{1t}, \gamma_{2t}]$ are the summations of the even and odd terms of $\boldsymbol{\alpha}_t$ respectively, and $[\gamma_{3t}, \cdots, \gamma_{nt}]$ are the second differences of $\boldsymbol{\alpha}_t$, it is nature to assume that their disturbances are independent to each other, making $\mathbf{Q}$ a diagonal matrix denoted by as $\text{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_n^2)$, in which $\sigma_1^2,\ \sigma_2^2$ are the variances of $w_{1t},\ w_{2t}$, and $\sigma_3^2, \cdots, \sigma_n^2$ are the variances of $w_{3t}, \cdots, w_{nt}$. For simplicity, we assume that σ_1^2 and σ_2^2 have the same value, denoted as σ_α^2, and all the remaining $\sigma_3^2, \cdots, \sigma_n^2$ are equal, their value denoted as σ_ϵ^2.

According the previous analysis, requiring $\sigma_\alpha^2 \gg \sigma_\epsilon^2$ for the new state vector is effectively forcing the second order difference of the original state $\boldsymbol{\alpha}_t$ to be small and thus imposing the smoothness constraint onto the estimated density curves. We can still use Algorithm 2 to update $\hat{\boldsymbol{\gamma}}_t$ after replacing $\mathbf{B}$ and $\boldsymbol{\alpha}_t$ with $\mathbf{BC}$ and $\boldsymbol{\gamma}_t$, respectively.

4.5 Parameter Estimation and Selection

In order for our prospective analysis to work, we do need a short starting up period, which is to gather a limited amount of training video data to initialize the parameters in the model. We typically use the first few hundreds of frames for parameter estimation, equivalent to the first 15 to 20 seconds of the process.

In our state-space model, there are two parameters σ_α^2 and σ_ϵ^2 that need to be estimated using the training data from the short starting up period, from $t = 1$ until time T. The two parameters determine the covariance matrix of the disturbance vector $\mathbf{w}_t$: σ_α^2 represents the degree of variability of the underlying state $\boldsymbol{\gamma}_t$, whereas σ_ϵ^2 controls its second order, indicating the smoothness of the estimated density curve. It is not convenient to find the values of σ_α^2 and σ_ϵ^2 by maximized likelihood estimation (MLE) as calculation of the likelihood of such a mixed system needs complicated process like importance sampling [111] and simulation smoothing [112], let alone to optimize the likelihood to estimate its parameters. Here, we adopt a Bayesian way to obtain the two parameters in the covariance matrix.

4.5.1 Bayesian Modeling and Sampling

We regard σ_α^2 and σ_ϵ^2 as latent random variables and choose their prior distributions, and then, obtain their posterior distribution through a sampling method and use the corresponding posterior means as the estimate of the parameters.

Since σ_α^2 and σ_ϵ^2 define the covariance matrix of the disturbance vector $\mathbf{w}_t$, which we assume follow a normal distribution, we choose the corresponding conjugate prior—an

inverse-gamma distribution, making the posterior distribution in the same family. We can write the hierarchical structure of the Bayesian model as:

$$\begin{aligned}
&Y_{it} \sim \text{Poisson}\{(\exp[\mathbf{BC}\boldsymbol{\gamma}_t])_i\}, \\
&\boldsymbol{\gamma}_t - \boldsymbol{\gamma}_{t-1} = \mathbf{w}_t \sim \text{normal}(\mathbf{0}, \mathbf{Q}),\ \mathbf{Q} = \text{diag}(\sigma_\alpha^2, \sigma_\alpha^2, \sigma_\epsilon^2, \cdots, \sigma_\epsilon^2), \\
&\sigma_\alpha^2 \sim \text{inverse-gamma}(a_1, b_1),\ \sigma_\epsilon^2 \sim \text{inverse-gamma}(a_2, b_2).
\end{aligned} \tag{4.13}$$

Compared to the original state-space model, the hierarchical model adds another layer associated with the prior distributions σ_α^2 and σ_ϵ^2, respectively. Once observing $\mathbf{Y}_1, \cdots, \mathbf{Y}_T$ in the starting up period, we employ a Markov chain Monte Carlo (MCMC) sampling method to update the posterior distributions of σ_α^2 and σ_ϵ^2, and then use the posterior means as the estimate of the two parameters.

Denote the values in the kth iteration of MCMC by $\boldsymbol{\gamma}_1^{(k)}, \cdots, \boldsymbol{\gamma}_T^{(k)}$, $(\sigma_\alpha^2)^{(k)}$ and $(\sigma_\epsilon^2)^{(k)}$. After the initialization, we sample $(\sigma_\alpha^2)^{(k)}$ and $(\sigma_\epsilon^2)^{(k)}$ through the Gibbs sampling, given $\boldsymbol{\gamma}_1^{(k-1)}, \cdots, \boldsymbol{\gamma}_T^{(k-1)}$. Since we adopt the conjugate priors, the posterior distributions are still inverse-gamma as:

$$(\sigma_\alpha^2)^{(k)} \sim \text{inverse-gamma}(a_1^{\text{post}}, b_1^{\text{post}}),\ (\sigma_\epsilon^2)^{(k)} \sim \text{inverse-gamma}(a_2^{\text{post}}, b_2^{\text{post}}), \tag{4.14}$$

where a_1^{post}, b_1^{post}, a_2^{post} and b_2^{post} are determined by a_1, b_1, a_2, b_2, and the sampled $\boldsymbol{\gamma}_t^{(k-1)}$. The derivation of the posterior distribution of $(\sigma_\alpha^2)^{(k)}$ and $(\sigma_\epsilon^2)^{(k)}$ is included in Appendix A.4.

Then, we sample $\boldsymbol{\gamma}_1^{(k)}, \cdots, \boldsymbol{\gamma}_T^{(k)}$, given $\mathbf{Q}^{(k)} = \text{diag}[(\sigma_\alpha^2)^{(k)}, (\sigma_\alpha^2)^{(k)}, (\sigma_\epsilon^2)^{(k)}, \cdots, (\sigma_\epsilon^2)^{(k)}]$ and the observations, $\mathbf{Y}_1, \cdots, \mathbf{Y}_T$. Unfortunately, the posterior distributions of $\boldsymbol{\gamma}_t^{(k)}$ are not of a standard type. We therefore implement a Metropolis-Hastings algorithm to sample $\boldsymbol{\gamma}_t^{(k)}$ from $t = 1$ to T. For each individual t, we first draw $\boldsymbol{\gamma}_t^{(k)}$ from the following

Algorithm 3 parameter estimation through Bayesian sampling

1. Initialize $\boldsymbol{\gamma}_1^{(0)}, \cdots, \boldsymbol{\gamma}_T^{(0)}$, $(\sigma_\alpha^2)^{(0)}$ and $(\sigma_\epsilon^2)^{(0)}$.
2. Set $k = 1$, then sample $(\sigma_\alpha^2)^{(k)}$ and $(\sigma_\epsilon^2)^{(k)}$ as: $(\sigma_\alpha^2)^{(k)} \sim$ inverse-gamma$(a_1^{\text{post}}, b_1^{\text{post}})$ and $(\sigma_\epsilon^2)^{(k)} \sim$ inverse-gamma$(a_2^{\text{post}}, b_2^{\text{post}})$, where

$$a_1^{\text{post}} = a_1 + (T-1), b_1^{\text{post}} = b_1 + \tfrac{1}{2}\sum_{j=1}^{2}\sum_{t=2}^{T}[\gamma_{jt}^{(k-1)} - \gamma_{j(t-1)}^{(k-1)}]^2,$$
$$a_2^{\text{post}} = a_2 + \tfrac{n-2}{2}(T-1), b_2^{\text{post}} = b_2 + \tfrac{1}{2}\sum_{j=3}^{n}\sum_{t=2}^{T}[\gamma_{jt}^{(k-1)} - \gamma_{j(t-1)}^{(k-1)}]^2.$$

3. Let $\mathbf{Q}^{(k)} = \text{diag}[(\sigma_\alpha^2)^{(k)}, (\sigma_\alpha^2)^{(k)}, (\sigma_\epsilon^2)^{(k)}, \cdots, (\sigma_\epsilon^2)^{(k)}]$.
4. Set $t = 1$, sample $\boldsymbol{\gamma}_t^{(k)}$ from a proposal distribution: $\boldsymbol{\gamma}_t^{(k)} \sim$ normal$(\boldsymbol{\gamma}_t^{(k-1)}, \mathbf{R})$.
5. Calculate the acceptance rate r as:

$$r = \frac{\prod_{i=1}^{m} p_{\text{poi}}(Y_{it}|[\mathbf{B}\mathbf{C}\boldsymbol{\gamma}_t^{(k)}]_i) p_{\text{nor}}(\boldsymbol{\gamma}_t^{(k)}|\boldsymbol{\gamma}_{t-1}^{(k)}, \mathbf{Q}^{(k)}) p_{\text{nor}}(\boldsymbol{\gamma}_t^{(k)}|\boldsymbol{\gamma}_{t+1}^{(k-1)}, \mathbf{Q}^{(k)})}{\prod_{i=1}^{m} p_{\text{poi}}(Y_{it}|[\mathbf{B}\mathbf{C}\boldsymbol{\gamma}_t^{(k-1)}]_i) p_{\text{nor}}(\boldsymbol{\gamma}_t^{(k-1)}|\boldsymbol{\gamma}_{t-1}^{(k)}, \mathbf{Q}^{(k)}) p_{\text{nor}}(\boldsymbol{\gamma}_t^{(k-1)}|\boldsymbol{\gamma}_{t+1}^{(k-1)}, \mathbf{Q}^{(k)})},$$

   where $p_{\text{poi}}(\cdot|\cdot)$ is the PDF of a Poisson distribution and $p_{\text{nor}}(\cdot|\cdot,\cdot)$ is the PDF of a multivariate normal distribution.
6. Generate a uniform random number, u, in $[0, 1]$. If $r > u$, accept $\boldsymbol{\gamma}_t^{(k)}$; otherwise set $\boldsymbol{\gamma}_t^{(k)} = \boldsymbol{\gamma}_t^{(k-1)}$.
7. Set $t = t + 1$, and repeat Step 4 to 6 until $t = T$.
8. Set $k = k + 1$, and repeat Step 2 to 7 until $k = K$.
9. Estimate σ_α^2 and σ_ϵ^2 as the posterior means:

$$\hat{\sigma}_\alpha^2 = \frac{1}{K - K_B}\sum_{k=K_B+1}^{K}(\sigma_\alpha^2)^{(k)}, \ \hat{\sigma}_\epsilon^2 = \frac{1}{K - K_B}\sum_{k=K_B+1}^{K}(\sigma_\epsilon^2)^{(k)}.$$

proposal distribution:

$$\boldsymbol{\gamma}_t^{(k)} \sim \text{normal}(\boldsymbol{\gamma}_t^{(k-1)}, \mathbf{R}), \tag{4.15}$$

where $\mathbf{R} = \text{diag}(\sigma_1^2, \sigma_1^2, \sigma_2^2, \cdots, \sigma_2^2)$ shares a similar structure as $\mathbf{Q}$. The acceptance ratio of a newly sampled $\boldsymbol{\gamma}_t^{(k)}$, r, is defined in a standard way, as the ratio of the conditional PDF given the current $\boldsymbol{\gamma}_t^{(k)}$ to that given the previous $\boldsymbol{\gamma}_t^{(k-1)}$. After getting r, we compare it with a uniform random variable, u, in $[0, 1]$, to determine whether to accept the new $\boldsymbol{\gamma}_t^{(k)}$ or not.

After repeating the above sampling iterations K times, the posterior means can be obtained by:

$$\hat{\sigma}_\alpha^2 = \frac{1}{K - K_B} \sum_{k=K_B+1}^{K} (\sigma_\alpha^2)^{(k)}, \; \hat{\sigma}_\epsilon^2 = \frac{1}{K - K_B} \sum_{k=K_B+1}^{K} (\sigma_\epsilon^2)^{(k)}, \tag{4.16}$$

where K_B is the amount of the burning steps. We list the detailed steps in Algorithm 3.

4.5.2 Select the Hyper-Parameters

In this subsection, we discuss the choices of the hyper-parameters in the Bayesian model (4.13) and the MCMC algorithm: a_1, b_1, a_2 and b_2 in the prior distribution, the initial values of the MCMC sampling, $\boldsymbol{\gamma}_t^{(0)}$, $(\sigma_\alpha^2)^{(0)}$ and $(\sigma_\epsilon^2)^{(0)}$; and σ_1^2 and σ_2^2 in the covariance matrix $\mathbf{R}$ of the proposal distribution. The parameters in the MCMC sampling matter less, as a long burning stage (namely a large enough K_B) will make the MCMC robust to initialization. As long as the MCMC has a good mixing, different proposal distributions give similar estimation results. We set those parameters in the following way: $(\sigma_\alpha^2)^{(0)} = 4 \times 10^{-2}$, $(\sigma_\epsilon^2)^{(0)} = 8 \times 10^{-4}$, run the modified Kalman filter in Algorithm 2 to obtain $\boldsymbol{\gamma}_t^{(0)}$, and let $\sigma_1^2 = 2 \times 10^{-2}$ and $\sigma_2^2 = 4 \times 10^{-4}$.

To determine the hyper-parameters in the inverse-gamma distributions of σ_α^2 and σ_ϵ^2, we follow Gelman [113]'s suggestion, which is to choose the non-informative prior as

Table 4.1: The parameters, $\hat{\sigma}_\alpha^2$ and $\hat{\sigma}_\epsilon^2$, estimated using the Video 1 data and under different b_2 values. In the following, $a_1 = a_2 = b_1 = 1.0$.

b_2	b_1/b_2	$\hat{\sigma}_\alpha^2$	$\hat{\sigma}_\epsilon^2$	$\hat{\sigma}_\alpha^2/\hat{\sigma}_\epsilon^2$
0.1	10	6.86×10^{-2}	4.17×10^{-3}	16.43
0.05	20	6.57×10^{-2}	3.78×10^{-3}	17.39
0.01	100	6.45×10^{-2}	3.62×10^{-3}	17.82
0.005	200	6.53×10^{-2}	3.70×10^{-3}	17.64

$a_1 = 1.0$ and $b_1 = 1.0$ for σ_α^2. To make sure $\sigma_\alpha^2 \gg \sigma_\epsilon^2$, we choose the same shape parameter $a_2 = 1.0$ but a much smaller scale parameter b_2 for σ_ϵ^2, as the mean of the inverse-gamma distribution is proportional to the scale parameter. We found that as long as b_1/b_2 is large enough, say, more than an order of magnitude, the estimation outcome appears robust. Table 4.1 presents the posterior means of the two parameters estimated from Video 1, with a total of $K = 9 \times 10^4$ iterations and $K_B = 3 \times 10^4$ burning steps. The estimated results are similar, despite a significant change in b_2. In practice, we recommend fixing $b_2 = 0.01$ as the default setting.

4.6 Application to TEM Videos

We test our state-space model and the online updating on the three clips of *in situ* TEM video described in Section 4.3. The number of the B-spline basis functions is fixed at 20 in all three cases. Because of incorporation of the smoothness constraint in our state-space model, our final estimation of the NPSD is not sensitive to the choices of the parameter. To save space, we discuss the full results on Video 1 clip. For the other two clips, we present limited analysis results to confirm the generality of the modeling and analysis.

4.6.1 Analysis of the Three Videos

Our first step is to find σ_α^2 and σ_ϵ^2 for each clip of videos. In Video 1, there are $1,149$ frames in total with 15 frame per second (fps) frame rate. We choose the first 300 frames

as the training set, corresponding to the first 20 seconds of the process. Using the Bayesian estimation method in Section 4.5.2 with the default parameter setting, we obtain the two parameters in the system as $\hat{\sigma}_{\alpha}^2 = 6.45 \times 10^{-2}$ and $\hat{\sigma}_{\epsilon}^2 = 3.62 \times 10^{-3}$.

Next we apply our updating method to the whole video. In our test, the TEM videos have already been fully recorded. We are mimicking a prospective analysis, starting at the end of the starting up period. For the remaining 849 frames in Video 1, the total processing time of using our algorithm is 1.23 second, or 1.5×10^{-4} seconds per frame, much faster than the frame rate of video (which is 15 frames per second or 0.067 seconds per frame). Combined with the image processing time (0.04 seconds per frame), the overall model processing is still fast enough for online monitoring. Figure 4.5 illustrates the updating process running from 25.67 second through 28.33 second. The upper row shows the input histograms, whereas the lower row shows the updated NPSDs. To demonstrate the difference of the estimated distributions, the time difference between two consecutive images in that plot is chosen to be 10 frames. It is evident that our Kalman filter updates the estimation of the time-varying NPSD with both the curve smoothness and temporal continuity.

We also show in Figure 4.6 the estimated NPSDs in different growth stages at 15s, 30s, 45s and 60s, respectively. Figure 4.6(a) presents the NPSD at the beginning of the growth stage when the nanocrystals are initializing in the chemical solution. The variance of the particle sizes is large and the support of the distribution is broad. Figure 4.6(b) presents a NPSD at the orientated attachment [83] growth stage, at which time the smaller particles collide with each other and are merged into larger ones. The variance of the particle sizes is smaller than that of the first stage. There is a noticeable bimodal pattern in the NPSD, in which the two peaks correspond to the sizes of the smaller particles and the merged (larger) particles, respectively. The final two plots in Figure 4.6 (c) and (d) are in the final growth stage, known as the Ostwald ripening [82] stage. In that stage, the larger

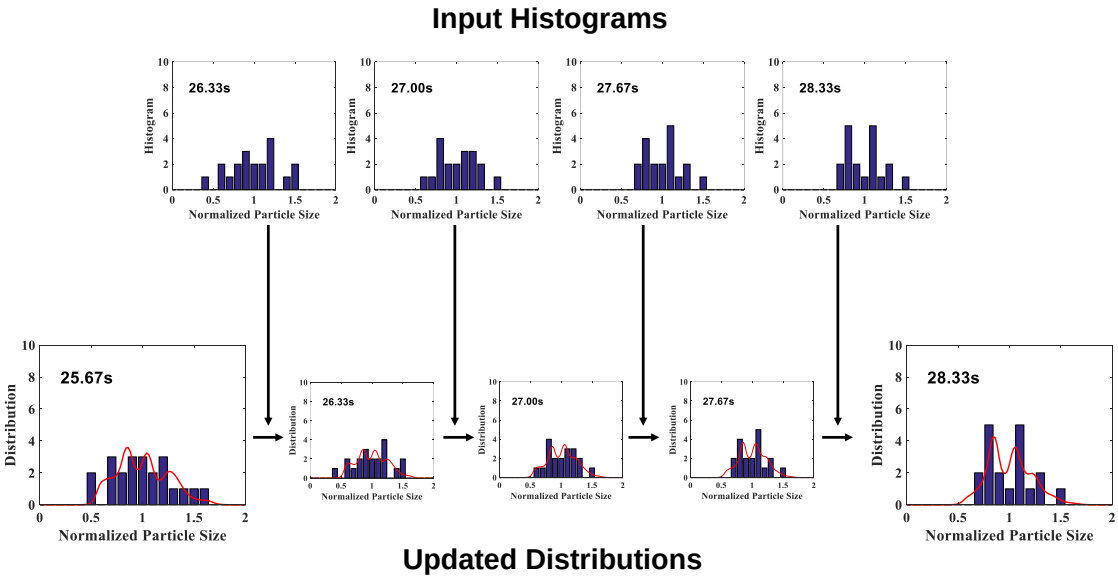


Figure 4.5: Illustration of the updating process of our state-space model.

particles grow at the expense of dissolving smaller particles. The size distribution tends to get concentrated and become unimodal. The variance continues to decrease. Material scientists expect to get nanocrystals having more uniform sizes at the end of the growth process. Our state-space model's online tracking results are consistent with the manual analysis results presented in the original report [1].

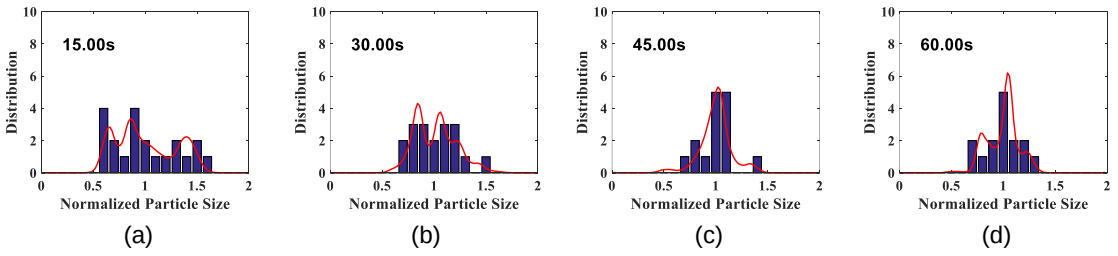


Figure 4.6: The estimated NPSD of Video 1 at different growth stages.

The last part of analysis performed on Video 1 is to show the innovation sequence of this nanocrystal growth process. Loosely speaking, the innovation sequence is the difference between what is newly observed at time t and what is anticipated, based on the state-space model and historical observations. In the past use of Kalman filter, the innovation sequence is commonly used to indicate a process change: if the underlying process is stable, then the innovation is supposedly to be random noise, whereas if the underlying process is going through a change, then the innovation sequence shows departure from random noise. The innovation at time t, denoted by $\boldsymbol{\nu}_t$ and its covariance matrix $\mathbf{F}_t$, is computed in Step 6 of Algorithm 2. To monitor the multivariate vector $\boldsymbol{\nu}_t$, we calculate the Mahalanobis squared distance [114] between $\boldsymbol{\nu}_t$ and $\mathbf{0}$ at each t, such that

$$A_t = \boldsymbol{\nu}_t^T \mathbf{F}_t^- \boldsymbol{\nu}_t. \tag{4.17}$$

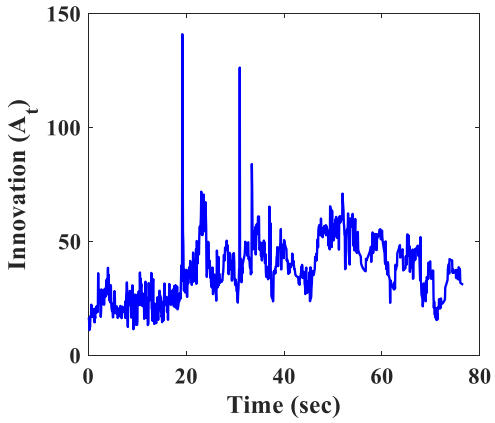


Figure 4.7: A_t obtained from the innovation sequence of the Kalman filter.

The sequence $\{A_1, A_2, \ldots, A_T\}$ for Video 1 is plotted in Figure 4.7. We observe that there is a transition period between the 20 second and 40 second time marks, and before

and after the transition period, the innovation sequence appears to have smaller magnitudes. This observation is consistent with the physical understanding discovered by Zheng et al. [1], i.e., the beginning stage of the growth is driven by the mechanism of orientated attachment, the latter stage is driven by the mechanism of Ostwald ripening, and there is a transition period in between. The timing of the transition period, discovered in the retrospective analysis [103], is between 25.8 second and 39.9 second. The result in Figure 4.7 shows that by tracking the innovation sequence of the state-space model, it offers the opportunity to detect the possible mechanism changes in the process.

Next, we test our algorithm on Video 2, which was published in the same paper [1] as Video 1 and captures a similar nanocrystal self-assembly growth process. There are total 637 frames in Video 2 with 15 fps frame rate. We still choose the first 300 frames to estimate the parameters. The Bayesian method produces the estimate of σ_α^2 as 8.15×10^{-2} and that of σ_ϵ^2 as 3.73×10^{-3}. Using these parameters, we estimate the NPSDs and show some results in Figure 4.8. The total updating time is 0.098 seconds, or 1.54×10^{-4} seconds per frame; this computational performance is consistent with that in processing Video 1 (and the image processing also costs 0.04 seconds per frame). Video 2 is a shorter clip and contains fewer particles. By observing the density plots in Figure 4.8, we are satisfied with the density curves estimated by our state-space model.

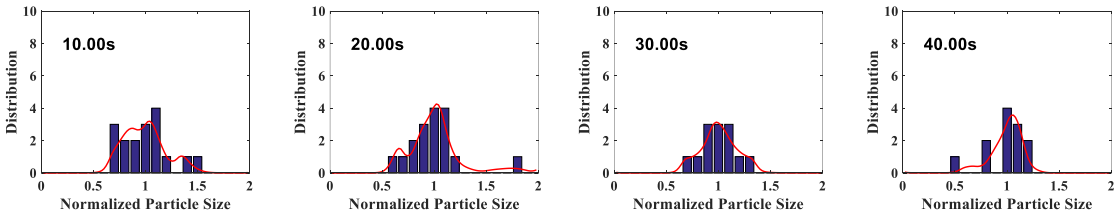


Figure 4.8: The estimated NPSDs of Video 2.

Lastly, we test our algorithm on Video 3. It was published in [2] and captures a different growth process than that in Videos 1 and 2. This process is of silver nanocrystal growth. There are only 112 frames in this video clip with 1 fps frame rate, so we pick the first 50 frames as the training set to estimate the parameters. For the process in Video 3, the parameters are accordingly estimated as $\sigma_\alpha^2 = 1.87 \times 10^{-1}$ and $\sigma_\epsilon^2 = 7.86 \times 10^{-3}$ by the proposed Bayesian method. Applying our updating method to Video 3, the total run time is 0.02 seconds, or 1.79×10^{-4} seconds per frame. The images processing time for Video 3 is 0.02 seconds per frame, so that the combined computation is again faster than the frame rate. Figure 4.9 presents the estimated NPSD of Video 3. In this process, the NPSD is always unimodal and its variance gets larger in the process.

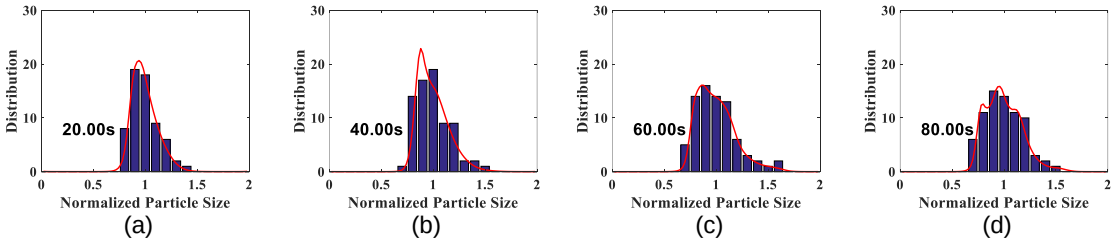


Figure 4.9: The estimated NPSDs of Video 3.

4.6.2 Comparison with Alternative Methods

In this section, we demonstrate the merits of the proposed method, especially the benefit of having both the curve smoothness and temporal continuity. We demonstrate all comparison results using Video 1 but the same insight holds true for other videos. We do not compare our method with a retrospective method because a retrospective (off-line) method sees all data and has the luxury of time, whereas a prospective (online) method

only sees a subset of the data, unless it reaches the very end of the video, and must be time conscious.

The first comparison is to conduct an out-of-sample quantitative test, comparing the proposed state-space method with three types of alternative: the first type is a pure histogram-based treatment (no smoothness constraint at all), the second type is to impose the curve smoothness within a frame but estimate the NPSD one frame a time without considering and imposing temporal continuity, and the third type is a state-space model without the curve smoothness (i.e., with temporal continuity across frames but no curve smoothness within a frame). In the second type of alternative, we include three popular methods: the smoothed histogram [101], the kernel estimation [17], and the penalized B-splines [18]. For the state-space model without the curve smoothness, we use $\boldsymbol{\alpha}_t$ instead of $\boldsymbol{\gamma}_t$ as the state, and the covariance matrix $\mathbf{Q}$ of the disturbance vector $\mathbf{w}_t$ is set as diag$(\sigma_\alpha^2, \sigma_\alpha^2, \cdots, \sigma_\alpha^2)$. The single parameter σ_α^2 can be estimated by a simplified Bayesian model, assuming $\sigma_\alpha^2 \sim$ inverse-gamma$(1, 1)$. The first 300 frames are still used for the training purpose. The Bayesian estimate of σ_α is 0.059, which is very close to that estimated in the previous subsection.

The out-of-sample test is to calculate the log-likelihood of the estimated probability density functions based on a number of observed nanocrystals. We randomly pick 90% the observed nanocrystals in each and every image frame and use them to establish our model and estimate the NPSD. Then, we use the remaining 10% observed nanocrystals in each and every frame to calculate the log-likelihood. For a given testing nanocrystal observation having a normalized particle size x^ℓ at time frame t, its log-likelihood is:

$$\log p_t(x^\ell) = \sum_{j=1}^{n} B_j(x^\ell)[\mathbf{C}\boldsymbol{\gamma}_t]_j - \log \text{ const}_t, \tag{4.18}$$

where const$_t$ is a normalization constant making the estimated NPSDs integrate to one.

Here $p_t(\cdot)$ is used for a probability density function, in order to be differentiated from the $f_t(\cdot)$ notation used earlier, which is an un-normalized density.

We then proceed to calculate the summation of the log-likelihoods for all the 10% out-of-sample testing nanocrystals at all time frames and use this summation as the accuracy metric of the distribution estimation. We repeat the out-of-sample test 500 times for each of the six methods. The mean of the log-likelihoods results are summarized in Table 4.2.

Table 4.2: Comparison results of the out-of-sample test among six approaches: using the observed histograms directly, three estimation methods considering the curve smoothness only, the state-space method without the curve smoothness, and the proposed method; all tested on Video 1.

Methods		Mean of log-likelihoods
Observed histograms (no constraint)		$-\infty$
Curve smoothness only	Smoothed histograms	-41.6
	Kernel estimation	-24.4
	Penalized B-splines	-46.7
State-space model (with temporal continuity)	Without curve smoothness	129.8
	With curve smoothness	196.1

In the out-of-sample test, the shortcoming of using the histogram directly is highlighted—almost all the log-likelihoods obtained are negative infinity. When certain samples fall into an empty interval of the histogram (meaning that this interval does not have any training observations), the direct histogram method sets the likelihood of this testing sample as 0, causing the log-likelihood to be negative infinity.

The distribution estimation methods with the curve smoothness can overcome this negative infinity problem. However, these methods estimate the distribution from each frame independently, lacking the ability to borrow information across time frames. When the number of observations at each frame is not large enough, they fail to produce a quality

estimate, as evident by the poor results in the out-of-sample test.

By using the state-space transition equations, the two state-space methods incorporate the temporal continuity, allowing the estimators to borrow information from other image frames and leading to much better performances than the other alternatives.

Between the state-space models with and without the curve smoothness, the one with the curve smoothness produces a much higher log-likelihood measure. We conduct a statistical testing and see whether the log-likelihood difference between the two approaches is significant. A one-way ANOVA, in which the null-hypothesis is that the two log-likelihoods have the same mean, yields a p-value of 6×10^{-162}, which confirm that the difference is indeed significant.

Given the benefit of using the state-space framework demonstrated above, we hence set the focus of the next two comparisons to be between the two state-space models, with and without the curve smoothness.

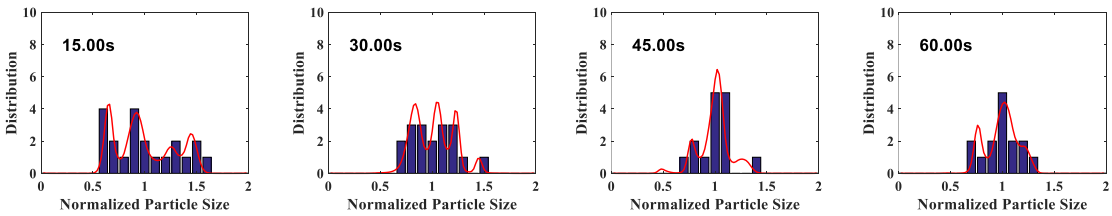


Figure 4.10: The estimated NPSDs of Video 1 by the state-space model without the curve smoothness.

The second comparison is to inspect the resulting NPSD obtained by the two state-space models. In Figure 4.10, we show the NPSDs, estimated at 15s, 30s, 45s and 60s by the state-space model without the curve smoothness. Comparing the results in Figure

4.6 obtained at the same time marks by the state-space model with the curve smoothness, the estimated distributions in Figure 4.10 are far worse, as they are very sensitive to small changes of any bin and do not handle well the existence of empty bins in a histogram. To see this point, consider the following observations. In Figure 4.10(b), while the orientated attachment growth mechanism suggests a bimodal distribution, the estimated distribution gives us three peaks. Between Figure 4.10(c) and (d), the variance is supposed to decrease, as this is in the Ostwald ripening growth stage, but the estimated distribution shows an increasing variance. When displaying the online distribution estimation frame by frame, it is obvious to us that the state-space model without the curve smoothness produces the time-varying NPSDs that are far more volatile and often react to noises and disturbances too dramatically.

The third comparison is to show the robustness of the proposed method to possible changes in the number of intervals in the input histograms. In the previous studies, we set the length of interval as 0.1 which gives 20 intervals in a histogram. In this comparison experiment, we test the cases by setting the length of interval to 0.2, 0.15, 0.08 and 0.05, respectively, and then estimate the corresponding NPSD, using the state-space model with and without the curve smoothness. We compare the resulting NPSDs with that obtained under the default setting, i.e., the length of a interval 0.1 or 20 intervals in the histogram. The difference between the two NPSDs is measured by a L_2 norm of the two density function curves.

In Figure 4.11(a), we plot the L_2-norm differences at each time frame between the NPSDs estimated, respectively, using the histograms with 10 intervals (the length of a interval 0.2) and 20 intervals (the length of a interval 0.1). It is apparent that inclusion of the curve smoothness leads to a much robust estimation outcome. In Figure 4.11(b), we present the summation over all frames of the L_2-norm differences between the NPSDs estimated, respectively, using histograms of a various number of intervals and the default

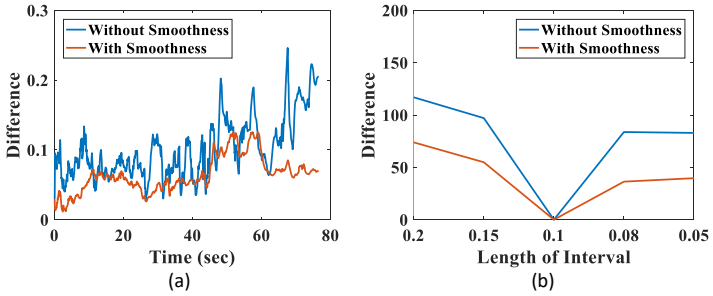


Figure 4.11: L_2-norm difference between two NPSDs: (a) L_2-norm differences at each time frame between the NPSDs estimated using the histograms with 10 intervals and 20 intervals; (b) the summation over all time frames of the L_2-norm differences between the NPSDs estimated using histograms of various lengths of intervals and the default setting.

setting (i.e., 20 intervals or interval length 0.1). In the broad range of choices, in which the default histogram parameter is doubled or halved (from 10 intervals to 40 intervals), the proposed method with the curve smoothness can get a rather robust estimation of the NPSD. By contrast, the state-space model without the curve smoothness performs comparably only in a much narrower range, roughly from using 16 intervals (0.125 interval length) to 22 intervals (0.09 interval length). This comparison demonstrates that by adding the curve smoothness, the proposed method is less sensitive to the parameter setting of the input histograms.

4.7 Summary

In this chapter, we propose an online method for monitoring the evolution of certain population characteristics observed in dynamic imaging (i.e., videos). Our model injects a flexible and robust modeling ability into a fast and closed-form updating algorithm. We demonstrate its application in monitoring the particle size distribution as a nanocrystal growth process is being observed by an *in situ* TEM.

The contributions of this work can be summarized as follows:

- The nonparametric method that models a time-varying probability density function and its specific tailoring to the nanocrystal growth process;
- A closed-form updating algorithm in the form of an extended Kalman filter for tracking the nanocrystal growth in real time;
- A sophisticated approach that addresses both modeling and estimation challenges, especially the incorporation of both the curve smoothness and temporal continuity in the modeling of a time-varying distribution.

Even though our method is demonstrated primarily in the context of estimating the normalized particle size distribution, we believe that the resulting method is general and should be applicable to other online distribution estimation problems. For other applications, one needs to replace the normalized particle size with a population characteristic of specific interest to that application. One importance assumption that may face challenges is the random walk assumption on the disturbance vector. Nonetheless, the random walk assumption appears a broadly accepted choice that can be a good starting point in a modeling effort, unless there exist contradicting evidences associated with a specific application to override its use.

5. CONCLUSIONS

In this chapter, we summarize the major contributions and impacts of the data science methods developed in this dissertation for handling nanoscale imaging data. We conclude this dissertation by discussing the possible extensions.

5.1 Summary

The contributions of the three chapters are listed as follows:

1. Chapter 2: We proposed a robust method to identify nanoparticles in low-quality TEM images. By combining two different kinds of image information, we improved the efficiency and accuracy of the nanoparticle detection. This laid the ground work for our subsequent research on *in situ* TEM videos, as the video frames are usually low-quality TEM images.

   This work tackles an important problem for nanoimaging data: how to exact accurate information from raw images or videos. As the volume of the original data is usually large and the quality is sometimes low, we cannot begin any further analysis before identifying and measuring nanoparticles accurately. This work also contributes to the general image processing area. It proposes a framework to improve detection accuracy by fusing two kinds of information in noisy and low-contrast images.

2. Chapter 3: We proposed a retrospective analysis for *in situ* TEM videos that identifies the change points in nanoparticle growth. This analysis objective was fulfilled by developing a new time-varying probability density function estimation method, followed by a robust change point detection procedure that is much less sensitive to the high degree of noises and stochasticity in *in situ* TEM videos.

   This work gives nanomaterial scientists a convenient tool to find interesting points

and periods from *in situ* TEM video clips that will save research time and energy and expedite scientific discoveries. Regarding methodology, this work also expands knowledge and generates new insights regarding how to handle the over-detection problem for change-point detection on time series data with high randomness.

3. Chapter 4: We proposed a prospective analysis for *in situ* TEM videos. We tracked and updated the growth status dynamically in a forward-looking way to identify the necessary control actions during growth. This analysis can be used to monitor and control the production of nanomaterials with desirable properties. An online, dynamic model such as this is the first of its kind.

   This work provides a possible way of designing an in-control self-assembly growth process with the help of *in situ* TEM video. As the online estimated particle size distribution can indicate the underlying growth mechanism, it can help us trigger the control action when the growth status is changing. Regarding methodology, our prospective analysis proposed a fast, closed-form method to estimate an online dynamic probability density function with the constraint of the curve smoothness and temporal continuity.

5.2 Future Study

Extending this dissertation's work may follow two directions: extending the current methods to solve other problems and developing new tools for emerging problems in the nanomanufacturing area.

Extend current methods: All the previous work, i.e., nanoparticle detection from TEM images and dynamic modeling of the nanoparticle growth process, can be extended to other applications. Accurate nanoparticle detection can help us identify microstructures in other TEM or SEM images. For example, in additive manufacturing, the surface of a product from a 3D printer usually needs to be polished. We are developing a method to

automate the polishing process and identifying particles on the surface from SEM images can determine the current polishing phase. The dynamic modeling of particles' behavior can also be applied to other applications. When objects of interest from video data are hard to track individually, the proposed dynamic modeling of the distribution can capture the overall changing trend. We are now investigating a video showing the performance of a sand remover on a solar panel. Our method can quantify the changing patterns of sand clusters lying on a solar panel and monitor the performance of its sand remover in real-time.

Develop new tools: When we have more and more data from advanced manufacturing, new problems emerge and new tools need to be developed. Two new needs draw our attentions. One is a fast, multi-shape analysis of nanoparticles from TEM images. The work in Chapter 2 is using an elliptical model to fit each particle, and it does not work well when particles have other shapes, like triangles or rectangles. There are some previous studies on multi-shape modeling based on non-parametric Bayesian method, but they are sensitive to noises and have high computational costs. We are developing a library-based method to overcome those drawbacks. It aims to learn a multiple shape library from training data, and when a new testing image comes, each particle will be identified according to the segmentation result and the closest prior shape in the library.

The other topic is to recover a high-resolution image from a low-resolution sample, which is referred to as super-resolution analysis in the field of image processing. It is not appropriate to apply existing super-resolution algorithms to SEM images directly, because some unique properties of those images, like a low-contrast level and blurred boundaries, are usually ignored in the generic super-resolution methods. We are working on methodologies to improve recovery accuracy and efficiency by considering those properties.

REFERENCES

[1] H. Zheng, R. K. Smith, Y. Jun, C. Kisielowski, U. Dahmen, and A. P. Alivisatos, "Observation of single colloidal platinum nanocrystal growth trajectories," *Science*, vol. 324, no. 5932, pp. 1309–1312, 2009.

[2] T. J. Woehl, C. Park, J. E. Evans, I. Arslan, W. D. Ristenpart, and N. D. Browning, "Direct observation of aggregative nanoparticle growth: Kinetic modeling of the size distribution and growth rate," *Nano Letters*, vol. 14, no. 1, pp. 373–378, 2013.

[3] M. Li, H. Schnablegger, and S. Mann, "Coupled synthesis and self-assembly of nanoparticles to give structures with controlled organization," *Nature*, vol. 402, no. 6760, pp. 393–395, 1999.

[4] A. K. Boal, F. Ilhan, J. E. DeRouchey, T. Thurn-Albrecht, T. P. Russell, and V. M. Rotello, "Self-assembly of nanoparticles into structured spherical and network aggregates," *Nature*, vol. 404, no. 6779, pp. 746–748, 2000.

[5] M. A. Boles, M. Engel, and D. V. Talapin, "Self-assembly of colloidal nanocrystals: From intricate structures to functional materials," *Chemical Reviews*, vol. 116, no. 18, pp. 11 220–11 289, 2016.

[6] ImageJ, "Image Processing and Analysis in Java, version 1.50d," 2015, available at http://rsbweb.nih.gov/ij/.

[7] C. Park, J. Huang, D. Huitink, S. Kundu, B. Mallick, H. Liang, and Y. Ding, "A multi-stage, semi-automated procedure for analyzing the morphology of nanoparticles," *IIE Transactions Special Issue on Nanomanufacturing*, vol. 44, no. 7, pp. 507–522, 2012.

[8] F. Yao, H. G. Müller, J. L. Wang *et al.*, "Functional linear regression analysis for longitudinal data," *The Annals of Statistics*, vol. 33, no. 6, pp. 2873–2903, 2005.

[9] C. Park, J. Huang, J. Ji, and Y. Ding, "Segmentation, inference and classification of partially overlapping nanoparticles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 669–681, 2013.

[10] C. Liu and D. B. Rubin, "The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence," *Biometrika*, vol. 81, no. 4, pp. 633–648, 1994.

[11] B. Konomi, S. S. Dahvala, J. Z. Huang, S. Kundu, D. Huitink, H. Liang, Y. Ding, and B. K. Mallick, "Bayesian object classification of gold nanoparticles," *The Annals of Applied Statistics*, vol. 7, no. 2, pp. 640–668, 2013.

[12] M. Jacobsen, *Point Process Theory and Applications: Marked Point and Piecewise Deterministic Processes*. Springer Science & Business Media, 2006.

[13] P. Pan and D. Schonfeld, "Video tracking based on sequential particle filtering on graphs," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1641–1651, 2011.

[14] W. Lu, J. Ting, J. J. Little, and K. P. Murphy, "Learning to track and identify players from broadcast sports videos," *IEEE Transactions on Pattern Analysis and Machine Intellifence*, vol. 35, no. 7, pp. 1704–1716, 2013.

[15] K. Park, L. F. Drummy, R. C. Wadams, H. Koerner, D. Nepal, L. Fabris, and R. A. Vaia, "Growth mechanism of gold nanorods," *Chemistry of Materials*, vol. 25, no. 4, pp. 555–563, 2013.

[16] C. Park, T. J. Woehl, J. E. Evans, and N. D. Browning, "Minimum cost multi-way data association for optimizing multitarget tracking of interacting objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 611–624, 2015.

[17] S. J. Sheather and M. C. Jones, "A reliable data-based bandwidth selection method for kernel density estimation," *Journal of the Royal Statistical Society, Series B*

(Methodological), vol. 53, no. 3, pp. 683–690, 1991.

[18] P. Eilers and B. Marx, "Flexible smoothing with b-splines and penalties," *Statistical Science*, vol. 11, no. 2, pp. 89–102, 1996.

[19] J. Ma, K. M. Kockelman, and P. Damien, "A multivariate Poisson-lognormal regression model for prediction of crash counts by severity using Bayesian methods," *Accident Analysis and Prevention*, vol. 40, no. 3, pp. 964–975, 2008.

[20] A. Rodriguez and E. Ter Horst, "Bayesian dynamic density estimation," *Bayesian Analysis*, vol. 3, no. 2, pp. 339–365, 2008.

[21] C. Zhang, N. Chen, and Z. Li, "State space modeling of autocorrelated multivariate Poisson counts," *IISE Transactions*, vol. 49, no. 5, pp. 518–531, 2017.

[22] R. H. Mena and M. Ruggiero, "Dynamic density estimation with diffusive Dirichlet mixtures," *Bernoulli*, vol. 22, no. 2, pp. 901–926, 2016.

[23] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613–624, 2001.

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[25] J. Watt, S. Cheong, and R. D. Tilley, "How to control the shape of metal nanostructures in organic solution phase synthesis for plasmonics and catalysis," *Nano Today*, vol. 8, no. 2, pp. 198–215, 2013.

[26] C. L. Nehl, H. Liao, and J. H. Hafner, "Optical properties of star-shaped gold nanoparticles," *Nano Letters*, vol. 6, no. 4, pp. 683–688, 2006.

[27] Y. Pan, S. Neuss, A. Leifert, M. Fischler, F. Wen, U. Simon, G. Schmid, W. Brandau, and W. Jahnen-Dechent, "Size-dependent cytotoxicity of gold nanoparticles," *Small*, vol. 3, no. 11, pp. 1941–1949, 2007.

[28] L. Shafarenko, M. Petrou, and J. Kittler, "Automatic watershed segmentation of randomly textured color images," *IEEE Transactions on Image Processing*, vol. 6, no. 11, pp. 1530–1544, 1997.

[29] Y. Lin, Y. Tsai, Y. Hung, and Z. Shih, "Comparison between immersion-based and toboggan-based watershed image segmentation," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 632–640, 2006.

[30] P. Quelhas, M. Marcuzzo, A. M. Mendonça, and A. Campilho, "Cell nuclei and cytoplasm joint segmentation using the sliding band filter," *IEEE Transactions on Medical Imaging*, vol. 29, no. 8, pp. 1463–1473, 2010.

[31] D. Freedman and T. Zhang, "Interactive graph cut based segmentation with shape priors," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 1, pp. 755–762, 2005.

[32] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[33] B. Parvin, Q. Yang, J. Han, H. Chang, B. Rydberg, and M. H. Barcellos-Hoff, "Iterative voting for inference of structural saliency and characterization of subcellular events," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 615–623, 2007.

[34] O. Schmitt and M. Hasse, "Morphological multiscale decomposition of connected regions with emphasis on cell clusters," *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 188–201, 2009.

[35] H. Yang and N. Ahuja, "Automatic segmentation of granular objects in images: Combining local density clustering and gradient-barrier watershed," *Pattern Recognition*, vol. 47, no. 6, pp. 2266–2279, 2014.

[36] P. J. De Temmerman, E. Verleysen, J. Lammertyn, and J. Mast, "Semi-automatic size measurement of primary particles in aggregated nanomaterials by transmission electron microscopy," *Powder Technology*, vol. 261, no. 1, pp. 191–200, 2014.

[37] P. Muneesawang and C. Sirisathitkul, "Size measurement of nanoparticle assembly using multilevel segmented TEM images," *Journal of Nanomaterials*, vol. 16, no. 1, pp. 58–63, 2015.

[38] L. L. Hench and J. K. West, "The sol-gel process," *Chemical Reviews*, vol. 90, no. 1, pp. 33–72, 1990.

[39] H. Zhang, H. Zhang, L. Tang, Z. Zhang, L. Gu, Y. Xu, and C. Eger, "Wear-resistant and transparent acrylate-based coating with highly filled nanosilica particles," *Tribology International*, vol. 43, no. 1, pp. 83–91, 2010.

[40] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration," *European Conference on Computer Vision*, vol. 2352, no. 1, pp. 408–422, 2002.

[41] X. Muñoz, J. Freixenet, X. Cufı, and J. Martı, "Strategies for image segmentation combining region and boundary information," *Pattern Recognition Letters*, vol. 24, no. 1, pp. 375–392, 2003.

[42] L. Chen, "Investigation on morphology measurement and evaluation of TiO_2 nanoparticles synthesized by sanss," *Journal of Alloys and Compounds*, vol. 483, no. 1, pp. 366–370, 2009.

[43] T. Pavlidis and Y.-T. Liow, "Integrating region growing and edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 3, pp. 225–233, 1990.

[44] M. Sato, S. Lakare, M. Wan, A. Kaufman, and M. Nakajima, "A gradient magnitude based region growing algorithm for accurate segmentation," *International Conference on Image Processing*, vol. 3, no. 1, pp. 448–451, 2000.

[45] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[46] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions

and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[47] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[48] C. Li, C. Y. Kao, J. C. Gore, and Z. Ding, "Minimization of region-scalable fitting energy for image segmentation," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1940–1949, 2008.

[49] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[50] M. S. Allili and D. Ziou, "Globally adaptive region information for automatic color–texture image segmentation," *Pattern Recognition Letters*, vol. 28, no. 15, pp. 1946–1956, 2007.

[51] K. Zhang, L. Zhang, H. Song, and W. Zhou, "Active contours with selective local or global segmentation: A new formulation and level set method," *Image and Vision Computing*, vol. 28, no. 4, pp. 668–676, 2010.

[52] Y. Tian, F. Duan, M. Zhou, and Z. Wu, "Active contour model combining region and edge information," *Machine Vision and Applications*, vol. 24, no. 1, pp. 47–61, 2013.

[53] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[54] G. Perrin, X. Descombes, and J. Zerubia, "A marked point process model for tree crown extraction in plantations," *IEEE International Conference on Image Processing*, vol. 1, no. 1, pp. 661–664, 2005.

[55] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*. John Wiley, 2002.

[56] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*. McGraw-Hill, 1995.

[57] A. Likas, N. Vlassis, and J. J Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[58] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," *International Conference on Information and Knowledge Management*, vol. 1, no. 1, pp. 600–607, 2002.

[59] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[60] W. R Taylor, J. M Thornton, and W. G Turnell, "An ellipsoidal approximation of protein shape," *Journal of Molecular Graphics*, vol. 1, no. 2, pp. 30–38, 1983.

[61] F. Herrera and L. Martínez, "A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multi-expert decision-making," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 31, no. 2, pp. 227–234, 2001.

[62] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 260–280, 2008.

[63] R. Fisker, J. M. Carstensen, M. F. Hansen, F. Bødker, and S. Mørup, "Estimation of nanoparticle size distributions by image analysis," *Journal of Nanoparticle Research*, vol. 2, no. 3, pp. 267–277, 2000.

[64] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*. Academic Press, 1980.

[65] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[66] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers*, vol. 100, no. 9, pp. 917–922, 1977.

[67] H. Scheffe, *The Analysis of Variance*. John Wiley & Sons, 1999.

[68] F. A. Sadjadi, "Experimental design methodology: The scientific tool for performance evaluation," *Signal and Image Processing Systems Performance Evaluation*, vol. 1310, no. 1, pp. 100–107, 1990.

[69] ——, "Infrared target detection with probability density functions of wavelet transform subbands," *Applied Optics*, vol. 43, no. 2, pp. 315–323, 2004.

[70] S. Cheong, J. Watt, B. Ingham, M. Toney, and R. Tilley, "*In situ* and *ex situ* studies of platinum nanocrystals: Growth and evolution in solution," *Journal of the American Chemical Society*, vol. 131, no. 40, pp. 14 590–14 595, 2009.

[71] S. B. Simonsen, I. Chorkendorff, S. Dahl, M. Skoglundh, J. Sehested, and S. Helveg, "Direct observations of oxygen-induced platinum nanoparticle ripening studied by *in situ* TEM," *Journal of the American Chemical Society*, vol. 132, no. 23, pp. 7968–7975, 2010.

[72] J. E. Evans, K. L. Jungjohann, N. D. Browning, and I. Arslan, "Controlled growth of nanoparticles from solution with *in situ* liquid transmission electron microscopy," *Nano Letters*, vol. 11, no. 7, pp. 2809–2813, 2011.

[73] Y. Liu, K. Tai, and S. J. Dillon, "Growth kinetics and morphological evolution of ZnO precipitated from solution," *Chemistry of Materials*, vol. 25, no. 15, pp. 2927–2933, 2013.

[74] P. Muneesawang, C. Sirisathitkul, and Y. Sirisathitkul, "Multi-level segmentation procedure for measuring the size distribution of nanoparticles in transmission electron microscope images," *Science of Advanced Materials*, vol. 7, no. 4, pp. 769–783, 2015.

[75] F. Wang, V. Richards, S. Shields, and W. Buhro, "Kinetics and mechanisms of aggregative nanocrystal growth," *Chemistry of Materials*, vol. 26, no. 1, pp. 5–21, 2013.

[76] J. Zhang, F. Huang, and Z. Lin, "Progress of nanocrystalline growth kinetics based on oriented attachment," *Nanoscale*, vol. 2, no. 1, pp. 18–34, 2010.

[77] D. S. Sholl and R. T. Skodje, "Late-stage coarsening of adlayers by dynamic cluster coalescence," *Physica A: Statistical Mechanics and Its Applications*, vol. 231, no. 4, pp. 631–647, 1996.

[78] H. G. Yang and H. C. Zeng, "Preparation of hollow anatase TiO_2 nanospheres via ostwald ripening," *The Journal of Physical Chemistry B*, vol. 108, no. 11, pp. 3492–3495, 2004.

[79] C. Ribeiro, E. J. Lee, E. Longo, and E. R. Leite, "A kinetic model to describe nanocrystal growth by the oriented attachment mechanism," *ChemPhysChem*, vol. 6, no. 4, pp. 690–696, 2005.

[80] X. Xue, R. L. Penn, E. R. Leite, F. Huang, and Z. Lin, "Crystal growth by oriented attachment: kinetic models and control factors," *CrystEngComm*, vol. 16, no. 8, pp. 1419–1429, 2014.

[81] C. Park, "Estimating multiple pathways of object growth using nonlongitudinal image data," *Technometrics*, vol. 56, no. 2, pp. 186–199, 2014.

[82] I. Lifshitz and V. Slyozov, "The kinetics of precipitation from supersaturated solid solutions," *Journal of Physics and Chemistry of Solids*, vol. 19, no. 1, pp. 35–50, 1961.

[83] D. Aldous, "Deterministic and stochastic models for coalescence (aggregation and coagulation): A review of the mean-field theory for probabilists," *Bernoulli*, vol. 5, no. 1, pp. 3–48, 1999.

[84] A. Baldan, "Review progress in Ostwald ripening theories and their applications to nickel-base superalloys Part I: Ostwald ripening theories," *Journal of Materials Science*, vol. 37, no. 11, pp. 2171–2202, 2002.

[85] I. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[86] R. Killick, P. Fearnhead, and I. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.

[87] P. Fryzlewicz, "Wild binary segmentation for multiple change-point detection," *The Annals of Statistics*, vol. 42, no. 6, pp. 2243–2281, 2014.

[88] Y. Qian, J. Z. Huang, X. Li, and Y. Ding, "Robust nanoparticles detection from noisy background by fusing complementary image information," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5713–5726, 2016.

[89] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. Addison-Wesley Reading, 1973.

[90] K. D. Zamba and D. M. Hawkins, "A multivariate change-point model for statistical process control," *Technometrics*, vol. 48, no. 4, pp. 539–549, 2006.

[91] Y. C. Yao, "Estimating the number of change-points via Schwarz's criterion," *Statistics & Probability Letters*, vol. 6, no. 3, pp. 181–189, 1988.

[92] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[93] G. C. Runger, T. R. Wlllemain, and S. Prabhu, "Average run lengths for cusum control charts applied to residuals," *Communications in Statistics-Theory and Methods*, vol. 24, no. 1, pp. 273–282, 1995.

[94] C. Kleiber, K. Hornik, F. Leisch, and A. Zeileis, "`strucchange`: An R package for testing for structural change in linear regression models," *Journal of Statistical Software*, vol. 7, no. 2, pp. 1–38, 2002.

[95] J. Chen and A. K. Gupta, "On change point detection and estimation," *Communications in Statistics - Simulation and Computation*, vol. 30, no. 3, pp. 665–697, 2001.

[96] B. Bian, W. Xia, J. Du, J. Zhang, J. P. Liu, Z. Guo, and A. Yan, "Growth mechanisms

and size control of fept nanoparticles synthesized using $Fe(Co)_x$($x<5$)-oleylamine and platinum(II) acetylacetonate," *Nanoscale*, vol. 5, no. 6, pp. 2454–2459, 2013.

[97] P. W. Voorhees, "The theory of Ostwald ripening," *Journal of Statistical Physics*, vol. 38, no. 1-2, pp. 231–252, 1985.

[98] S. Hardy and P. Voorhees, "Ostwald ripening in a system with a high volume fraction of coarsening phase," *Metallurgical Transactions A*, vol. 19, no. 11, pp. 2713–2721, 1988.

[99] A. Lo and R. T. Skodje, "Kinetic and Monte Carlo models of thin film coarsening: Cross over from diffusion-coalescence to Ostwald growth modes," *The Journal of Chemical Physics*, vol. 112, no. 4, pp. 1966–1974, 2000.

[100] M. Grzelczak, J. Vermant, E. M. Furst, and L. M. Liz-Marzán, "Directed self-assembly of nanoparticles," *ACS Nano*, vol. 4, no. 7, pp. 3591–3605, 2010.

[101] J. S. Simonoff, "A penalty function approach to smoothing large sparse contingency tables," *The Annals of Statistics*, vol. 11, no. 1, pp. 208–218, 1983.

[102] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 4, no. 2, pp. 639–650, 1994.

[103] Y. Qian, J. Z. Huang, and Y. Ding, "Identifying multi-stage nanocrystal growth using *in situ* TEM video data," *IISE Transactions*, vol. 49, no. 5, pp. 532–543, 2017.

[104] L. Ljung, "Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems," *IEEE Transactions on Automatic Control*, vol. 24, no. 1, pp. 36–50, 1979.

[105] Y. Bishop, S. Fienberg, and P. Holland, *Discrete Multivariate Analysis: Theory and Practice*. Springer-Verlag, 1975.

[106] B. Jørgensen, S. Lundbye-Christensen, P. X. Song, and L. Sun, "A state space model for multivariate longitudinal count data," *Biometrika*, vol. 86, no. 1, pp. 169–181,

1999.

[107] R. C. Jung, R. Liesenfeld, and J. F. Richard, "Dynamic factor models for multivariate count data: An application to stock-market trading activity," *Journal of Business and Economic Statistics*, vol. 29, no. 1, pp. 73–85, 2011.

[108] S. L. Zeger, "A regression model for time series of counts," *Biometrika*, vol. 75, no. 4, pp. 621–629, 1988.

[109] D. Karlis and L. Meligkotsidou, "Multivariate Poisson regression with covariance structure," *Statistics and Computing*, vol. 15, no. 4, pp. 255–265, 2005.

[110] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[111] J. Durbin and S. J. Koopman, "Monte Carlo maximum likelihood estimation for non-Gaussian state space models," *Biometrika*, vol. 84, no. 3, pp. 669–684, 1997.

[112] P. de Jong and N. Shephard, "The simulation smoother for time series models," *Biometrika*, vol. 82, no. 2, pp. 339–350, 1995.

[113] A. Gelman, "Prior distributions for variance parameters in hierarchical models," *Bayesian Analysis*, vol. 1, no. 3, pp. 515–534, 2006.

[114] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences of India*, vol. 2, no. 1, pp. 49–55, 1936.

[115] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*. Oxford University Press, 2012.

[116] W. M. Bolstad and J. M. Curran, *Introduction to Bayesian Statistics*. John Wiley & Sons, 2016.

APPENDIX A

DERIVATIONS OF ALGORITHMS IN VIDEO ANALYSIS

A.1 Optimization of Density Estimation (Section 3.3 of Chapter 3)

We proposed to maximize the penalized log likelihood of the density functions by ADMM [24]. We write the corresponding augmented Lagrangian as:

$$\begin{aligned} L_\rho(\{a_{jt}\},\{z_{jt}\},\{c_{it}\}) = &\sum_{t=1}^{T} L_t(\{a_{jt}\}) - (\lambda_2/2)\sum_{j=1}^{n}\sum_{t=1}^{T-1}(\Delta_2 z_{jt})^2 \\ &- \rho\sum_{t=1}^{T}\sum_{j=1}^{n} c_{jt}(a_{jt}-z_{jt}) - (\rho/2)\sum_{t=1}^{T}\sum_{j=1}^{n}(a_{jt}-z_{jt})^2, \end{aligned} \tag{A.1}$$

where

$$L_t(\{a_{jt}\}) = \sum_{i=1}^{m} y_{it}\eta_{it} - \sum_{i=1}^{m}\exp(\eta_{it}) - \lambda_1\sum_{j=1}^{n-1}\frac{(\Delta_1 a_{jt})^2}{2}. \tag{A.2}$$

The ADMM algorithm targets to find the saddle point of Equation [3.8], defined as:

$$(\{\hat{a}_{jt}\},\{\hat{z}_{jt}\},\{\hat{c}_{it}\}) = \arg\min_{\{c_{it}\}}\max_{\{a_{jt}\},\{z_{jt}\}} L_\rho, \tag{A.3}$$

where $\{\hat{a}_{jt}\}$ will be the maximizer of the penalized log likelihood of the density functions.

The saddle point is found by using the coordinate decent method [89]. First we change the $\min$-$\max$ problem to a $\max$-$\min$ one by adding a negative sign in Equation (3.8) and rewrite it in a matrix form:

$$\begin{aligned} L'_\rho(A,Z,C) = &-\sum_{t=1}^{T} L_t(A) + (\lambda_2/2)\sum_{j=1}^{n}\sum_{t=1}^{T-1}(\Delta_2 z_{jt})^2 \\ &+ \rho C^T(A-Z) + (\rho/2)\sum_{t=1}^{T}||A-Z||_2^2, \end{aligned} \tag{A.4}$$

where $(A)_{jt} = a_{jt}$, $(Z)_{jt} = z_{jt}$ and $(C)_{jt} = c_{jt}$. Then we update A, Z, C iteratively to get the saddle point. When updating one of the three variables, we will fix the other two. The values of variables in the qth iteration are signified via the (q) superscript.

To update A, we solve:

$$\arg\min_{A} - \sum_{t=1}^{T} L_t(A) + (\rho/2)||A - Z^{(q)} + C^{(q)}||_2^2. \tag{A.5}$$

The problem can be decomposed into T independent subproblems for each time t. Denote the t-th column of A, Z and C by a_t, z_t and c_t, respectively. The difference operator Δ_1 can be rewritten as a matrix multiplication:

$$\sum_{j=1}^{n-1} (\Delta_1 a_{jt})^2 = a_t' D_1' D_1 a_t, \tag{A.6}$$

where D_1 is an $n \times n$ matrix with $(D_1)_{jj}$ as -1, $(D_1)_{j(j-1)}$ as 1, for $j = 2, \ldots, n$, and all other elements as 0.

We can update a_t by:

$$\begin{aligned} a_t^{(q+1)} = \arg\min_{a_t} \Big\{ & - \sum_{i=1}^{m} y_{it}\eta_{it} + \sum_{i=1}^{m} \exp(\eta_{it}) \\ & + (\lambda_1/2) a_t' D_1' D_1 a_t + (\rho/2)[a_t' a_t - 2(z_t^{(q)} - c_t^{(q)})' a_t] \Big\}. \end{aligned} \tag{A.7}$$

The solution of the above minimization problem can follow [18], as they solved a similar problem. But we need to make some modification to what [18] did because we included a new term (the fourth term in the large bracket) in the above objective function. According to [18], the solution of a_t is to set the first derivative of the above objective function to

zero. That leads us to the following equation:

$$B_t^+ - B' \exp(Ba_t) = \lambda_1 D_1' D_1 a_t + \rho[a_t - (z_t^{(q)} - c_t^{(q)})], \tag{A.8}$$

where $B_t^+ = [B_{1t}^+, \cdots, B_{nt}^+]'$ and $B_{jt}^+ = \sum_s B_j(\phi_{st})$. Unfortunately, the above equation does not have a closed-form solution for a_t, so we have to solve it through an iterative procedure by using the following equation (which did a first order Taylor expansion of the exponential term, so that a_t can be solved through a weighted linear regression):

$$\begin{aligned} & B_t^+ - B' \exp(B\hat{a}_t) + B'B\hat{a}_t + \rho(z_t^{(q)} - c_t^{(q)}) \\ = & [B'B + \lambda_1 D_1' D_1 + \rho I_n] a_t, \end{aligned} \tag{A.9}$$

where I_n is the $n \times n$ identity matrix and $\hat{a}_t$ is the result estimated from the previous iteration, whose initial value is set to be $a_t^{(q)}$ (from the q-th step). Once the numerical iterative procedure converges, the resulting a_t is treated as $a_t^{(q+1)}$.

To update Z, we solve:

$$\begin{aligned} Z^{(q+1)} = \arg\min_Z \Big\{ & \lambda_2 \sum_{j=1}^{n} \sum_{t=1}^{T-1} (\Delta_2 z_{jt})^2 \\ & + (\rho/2) ||A^{(q+1)} - Z + C^{(q)}||_2^2 \Big\}. \end{aligned} \tag{A.10}$$

The terms in the large bracket can be rewritten as:

$$||A^{(q+1)} - Z + C^{(q)}||_2^2 + \frac{\lambda_2}{\rho} \sum_{j=1}^{n} \sum_{t=1}^{T-1} (\Delta_2 z_{jt})^2. \tag{A.11}$$

The second term can be transformed into a matrix form:

$$\sum_{j=1}^{n} \sum_{t=1}^{T-1} (\Delta_2 z_{it})^2 = ||ZD_2||_2^2 \tag{A.12}$$

where D_2 is a $T \times T$ matrix with $(D_2)_{tt}$ as -1, $(D_2)_{t(t+1)}$ as 1, for $t = 1, \ldots, T-1$, and all other elements as 0. In fact, the above minimization problem has a closed-form solution for $Z^{(q+1)}$, which is

$$Z^{(q+1)} = \left[A^{(q+1)} + C^{(q)}\right] (I_T + \frac{\lambda_2}{\rho} D_2 D_2')^{(-1)}. \tag{A.13}$$

At last, we update the Lagrangian multipliers C by:

$$C^{(q+1)} = C^{(q)} + (A^{(q+1)} - Z^{(q+1)}). \tag{A.14}$$

We continue the iteration until all those variables converge.

A.2 Gaussian Approximation of Poisson Distribution (Section 4.4.2 of Chapter 4)

The Poisson distributed observation equation can be written as

$$Y_{it} \sim \text{Poisson}\{(\exp[\mathbf{B}\boldsymbol{\alpha}_t])_i\},\ i = 1, \ldots, m, \tag{A.15}$$

and we would like to find a Gaussian distribution

$$\mathbf{Y}_t \sim \text{Normal}(\mathbf{B}\boldsymbol{\alpha}_t + \boldsymbol{\mu}_t, \mathbf{H}_t), \tag{A.16}$$

to approximate it. Durbin and Koopman [111] proposed that if the probability distribution functions (PDFs) in Equation (A.15) and (A.16) have the same first and second derivatives w.r.t the state $\boldsymbol{\alpha}_t$, then Equation (A.16) can serve as a good approximation of Equation (A.15) in updating the state-space model. We can use this idea to calculate $\boldsymbol{\mu}_t$ and $\mathbf{H}_t$ in Equation (A.16). To simplify the derivation, we use $\mathbf{B}\boldsymbol{\alpha}_t$ instead of $\boldsymbol{\alpha}_t$ as the variable to calculate those derivatives.

The logarithm of the PDFs in Equation (A.15) and Equation (A.16), as a function of

$\mathbf{B}\boldsymbol{\alpha}_t$, can be expressed, respectively, as

$$\log p_p([\mathbf{B}\boldsymbol{\alpha}_t]_i) = Y_{it}[\mathbf{B}\boldsymbol{\alpha}_t]_i - \exp[\mathbf{B}\boldsymbol{\alpha}_t]_i, i = 1, \ldots, m, \tag{A.17}$$

and

$$\log p_g(\mathbf{B}\boldsymbol{\alpha}_t) = -\frac{1}{2}(\mathbf{Y}_t - \mathbf{B}\boldsymbol{\alpha}_t - \boldsymbol{\mu}_t)^T \mathbf{H}_t^{-1}(\mathbf{Y}_t - \mathbf{B}\boldsymbol{\alpha}_t - \boldsymbol{\mu}_t) + \text{const}, \tag{A.18}$$

where 'const' is a term unrelated to $\boldsymbol{\alpha}_t$.

In Equation (A.17), the PDF of each coordinate of $\mathbf{B}\boldsymbol{\alpha}_t$ is independent to each other, Equation (A.18) should have the same property, meaning that $\mathbf{H}_t$ should be a diagonal matrix. We can then rewrite Equation (A.18) as:

$$\log p_g([\mathbf{B}\boldsymbol{\alpha}_t]_i) = -\frac{1}{2[\mathbf{H}_t]_{ii}}(Y_{it} - [\mathbf{B}\boldsymbol{\alpha}_t]_i - [\boldsymbol{\mu}_t]_i)^2 + \text{const}, \tag{A.19}$$

Then calculating the first and second derivatives of Equation (A.17) and (A.19) w.r.t $[\mathbf{B}\boldsymbol{\alpha}_t]_i$ and equating them at the estimated $\hat{\boldsymbol{\alpha}}_t$, we get the following two equations:

$$Y_{it} - \exp[\mathbf{B}\hat{\boldsymbol{\alpha}}_t]_i = \frac{1}{[\mathbf{H}_t]_{ii}}(Y_{it} - [\mathbf{B}\hat{\boldsymbol{\alpha}}_t]_i - [\boldsymbol{\mu}_t]_i), \tag{A.20}$$

and

$$\exp[\mathbf{B}\hat{\boldsymbol{\alpha}}_t]_i = \frac{1}{[\mathbf{H}_t]_{ii}}. \tag{A.21}$$

The two equations further yield:

$$[\mathbf{H}_t]_{ii} = \frac{1}{\exp[\mathbf{B}\hat{\boldsymbol{\alpha}}_t]_i} = \exp[-\mathbf{B}\hat{\boldsymbol{\alpha}}_t]_i, \tag{A.22}$$

and

$$[\boldsymbol{\mu}_t]_i = Y_{it} - [\mathbf{B}\hat{\boldsymbol{\alpha}}_t]_i - \exp[-\mathbf{B}\hat{\boldsymbol{\alpha}}_t]_i(Y_{it} - \exp[\mathbf{B}\hat{\boldsymbol{\alpha}}_t]_i). \tag{A.23}$$

Rewriting Equation (A.22) and (A.23) in a matrix form, we finally obtain $\boldsymbol{\mu}_t$ and $\mathbf{H}$ as:

$$\begin{aligned} \boldsymbol{\mu}_t &= \mathbf{Y} - \mathbf{B}\hat{\boldsymbol{\alpha}}_t - \exp(-\mathbf{B}\hat{\boldsymbol{\alpha}}_t)[\mathbf{Y} - \exp(\mathbf{B}\hat{\boldsymbol{\alpha}}_t)], \\ \mathbf{H} &= \text{diag}(\exp(-\mathbf{B}\hat{\boldsymbol{\alpha}}_t)). \end{aligned} \tag{A.24}$$

A.3 Detailed Steps of Kalman Filter (Section 4.4.2 of Chapter 4)

Given a linear Gaussian state-space model

$$\begin{aligned} &\mathbf{Y}_t \sim \text{Normal}(\mathbf{B}\boldsymbol{\alpha}_t + \boldsymbol{\mu}_t, \mathbf{H}_t), \\ &\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t + \mathbf{w}_t,\ \mathbf{w}_t \sim \text{Normal}(\mathbf{0}, \mathbf{Q}), \end{aligned} \tag{A.25}$$

the Kalman filter can estimate the state $\boldsymbol{\alpha}_t$ in a recursive way from $t = 1$ to time T. First we need to predict $\boldsymbol{\alpha}_t$ and its covariance according to the estimation of the previous step as

$$\begin{aligned} \hat{\boldsymbol{\alpha}}_t^- &= \hat{\boldsymbol{\alpha}}_{t-1}, \\ \mathbf{P}_t^- &= \mathbf{P}_{t-1} + \mathbf{Q}, \end{aligned} \tag{A.26}$$

where $\hat{\boldsymbol{\alpha}}_t^-$ is called the prior estimator and $\mathbf{P}_t^-$ is the prior covariance matrix. The two equations above can be derived from the distribution of $p(\boldsymbol{\alpha}_t|\mathbf{Y}_1, \cdots, \mathbf{Y}_{t-1})$ [115].

When a new $\mathbf{Y}_t$ is coming, we calculate the innovation $\boldsymbol{\nu}_t$ and its covariance matrix according to the previous prediction $\hat{\boldsymbol{\alpha}}_t^-$ and the new input $\mathbf{Y}_t$ as

$$\begin{aligned} \boldsymbol{\nu}_t &= \mathbf{Y}_t - \mathbf{B}\hat{\boldsymbol{\alpha}}_t^- - \boldsymbol{\mu}_t; \\ \mathbf{F}_t &= \mathbf{B}\mathbf{P}_t^-\mathbf{B}^T + \mathbf{H}_t. \end{aligned} \tag{A.27}$$

Then the Kalman gain will be calculated as:

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{B}^T \mathbf{F}_t^{-1}. \tag{A.28}$$

At last we update the estimator of state $\boldsymbol{\alpha}_t$ and its covariance matrix as:

$$\begin{aligned} \hat{\boldsymbol{\alpha}}_t &= \hat{\boldsymbol{\alpha}}_t^- + \mathbf{K}_t \boldsymbol{\nu}_t, \\ \mathbf{P}_t &= \mathbf{P}_t^- (\mathbf{I} - \mathbf{K}_t \mathbf{B})^T, \end{aligned} \tag{A.29}$$

where $\hat{\boldsymbol{\alpha}}_t$ is called the posterior estimator and $\mathbf{P}_t^-$ is the posterior covariance matrix. Those equations can be derived from the distribution of $p(\boldsymbol{\alpha}_t | \mathbf{Y}_1, \cdots, \mathbf{Y}_t)$ [115].

A.4 Posterior Distribution of σ_α^2 and σ_ϵ^2 (Section 4.5.1 of Chapter 4)

Here we want to show the derivation of the posterior distribution of σ_α^2 and σ_ϵ^2 in our Bayesian model:

$$\begin{aligned} &Y_{it} \sim \text{Poisson}\{(\exp[\mathbf{B}\mathbf{C}\boldsymbol{\gamma}_t])_i\}, \\ &\boldsymbol{\gamma}_{t+1} - \boldsymbol{\gamma}_t = \mathbf{w}_t \sim \text{normal}(\mathbf{0}, \mathbf{Q}),\ \mathbf{Q} = \text{diag}(\sigma_\alpha^2, \sigma_\alpha^2, \sigma_\epsilon^2, \cdots, \sigma_\epsilon^2), \\ &\sigma_\alpha^2 \sim \text{inverse-gamma}(a_1, b_1),\ \sigma_\epsilon^2 \sim \text{inverse-gamma}(a_2, b_2). \end{aligned} \tag{A.30}$$

Since $\mathbf{Q}$ is a covariance matrix, we can rewrite the second layer of the model (4.13) as:

$$\gamma_{jt} - \gamma_{j(t-1)} \sim \text{normal}(0, \sigma_\alpha^2),\ j = 1, 2,\ t = 2, \cdots, T, \tag{A.31}$$

and

$$\gamma_{jt} - \gamma_{j(t-1)} \sim \text{normal}(0, \sigma_\epsilon^2),\ j = 3, \cdots, n,\ t = 2, \cdots, T. \tag{A.32}$$

For $j = 1, 2$, $\gamma_{jt} - \gamma_{j(t-1)}$ are regarded as $2(T-1)$ i.i.d variables following normal$(0, \sigma_\alpha^2)$.

Since we choose the conjugate prior $\sigma_\alpha^2 \sim$ inverse-gamma(a_1, b_1), its posterior distribution has the same formation as inverse-gamma$(a_1^{\text{post}}, b_1^{\text{post}})$. As derived in [116], a_1^{post} and b_1^{post} are calculated as:

$$a_1^{\text{post}} = a_1 + \frac{1}{2}2(T-1) = a_1 + (T-1), \tag{A.33}$$

and

$$b_1^{\text{post}} = b_1 + \frac{1}{2}\sum_{j=1}^{2}\sum_{t=2}^{T}[\gamma_{jt} - \gamma_{j(t-1)}]^2. \tag{A.34}$$

For $j = 3, \cdots, n$, $\gamma_{jt} - \gamma_{j(t-1)}$ are regarded as $(n-2)(T-1)$ i.i.d variables following normal$(0, \sigma_\epsilon^2)$. Following the same derivation, the posterior distribution of σ_ϵ^2 is written as inverse-gamma$(a_2^{\text{post}}, b_2^{\text{post}})$ with

$$a_2^{\text{post}} = a_2 + \frac{1}{2}(n-2)(T-1), \tag{A.35}$$

and

$$b_2^{\text{post}} = b_2 + \frac{1}{2}\sum_{j=3}^{n}\sum_{t=2}^{T}[\gamma_{jt} - \gamma_{j(t-1)}]^2. \tag{A.36}$$