HIGH DIMENSIONAL DATA ANALYSIS FOR ANOMALY DETECTION AND QUALITY IMPROVEMENT

A Dissertation
Presented to
The Academic Faculty

By

Hao Yan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology

August 2017

HIGH DIMENSIONAL DATA ANALYSIS FOR ANOMALY DETECTION AND QUALITY IMPROVEMENT

Approved by:

Dr. Jianjun Shi, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Kamran Paynabar, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Roshan Vengazhiyil
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Yajun Mei
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Massimo Ruzzene
D. Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: April 10, 2017

To my parents, my wife,

we together made this journey a lot memorable.

Hao Yan

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my advisors, Professor Jianjun Shi and Professor Kamran Paynabar, for their support, patience, and encouragement throughout my Ph.D. studies.

Professor Kamran Paynabar has taught me innumerable lessons and insights regarding to academic research, from creating research ideas and developing methodologies, to expressing contributions in the form of technical writing. I am grateful to him for holding me to a high research standard and enforcing strict validations for each research result. I would also like to express my gratitude to him for carefully reading and commenting on countless revisions of my research papers and presentations.

Professor Shi, has been always there to listen and give advices. Professor Shi has been supportive and has given me the freedom to pursue various projects. I am deeply grateful to him for his advices that helped me sort out the research ideas and the technical details of my work. He has given me many helpful advices not only on how to pursue my academic career path but also help me become more mature.

I would like to thank my committee members, Professor Roshan Vengazhiyil, Professor Yajun Mei, and Professor Massimo Ruzzene for their dedications and constructive suggestions on my dissertation.

I thank Dr. Kaibo Liu, Dr. Olivier Mesnil, Xiaowei Yue, Chen Zhang, Xiaolei Fang, and Mohammed Nabhan for their remarkable efforts in collaborating with me on different research papers and projects.

TABLE OF CONTENTS

Acknowledgments v

List of Tables x

List of Figures xi

Chapter 1: Introduction and Background 1

1.1 Research Background 1

1.2 Specific Research Topics 2

1.2.1 Anomaly Detection for High-dimensional Functional Profiles 2

1.2.2 Anomaly Detection for High-Dimensional (HD) Functional Data Streams with Smooth Spatial-temporal Structures 4

1.2.3 An Adaptive Framework for Online Sensing and Anomaly Detection 7

1.2.4 Modeling the High-dimensional Structured Point Cloud with Process Variables 8

1.3 Thesis Organization 9

Chapter 2: Literature Review 11

2.1 Monitoring and Diagnosis of High-dimensional Streaming Functional Data 14

2.2 Adaptive Sampling and Sensing for HD spatial profiles 15

2.3 Modeling Point Cloud data 16

2.4 Tensor-on-scalar regression 18

Chapter 3: Anomaly Detection for Images and High-dimensional Spatial Functional Profile 20

3.1 Smooth-Sparse Decomposition (SSD) 20

3.1.1 Optimization Algorithms for SSD 22

3.1.2 Generalization to 2-D image case 26

3.1.3 Choice of tuning parameters λ and γ 27

3.1.4 Choice of basis for background and anomalous regions 30

3.2 Simulation study 30

3.3 Case study 38

3.4 Conclusion 40

Chapter 4: Real-time Monitoring and Diagnosis of High-Dimensional Functional Data Streams via Spatio-Temporal Smooth Sparse Decomposition . 43

4.1 Spatio-Temporal Smooth Sparse Decomposition 44

4.2 ST-SSD for Streaming Data and Recursive Estimation 48

4.2.1 Reproducing Kernels 49

4.2.2 Roughness Minimization 50

4.2.3 ST-SSD for Stationary Processes 53

4.3 Online Process Monitoring and Diagnostics 54

4.3.1 Construct Monitoring Statistics 54

4.3.2 Control Limit Determination 55

4.3.3 Diagnosis of Detected Changes 56

4.4 Simulation Study 57

4.5 Case study . . . 64
4.5.1 On-line Seam Detection in Steel Rolling Process . . . 64
4.5.2 Online Monitoring of Solar Activity . . . 65
4.5.3 Tonnage Signal Monitoring . . . 67
4.6 Conclusion . . . 70

Chapter 5: An Adaptive Framework for Online Sensing and Anomaly Detection 72

5.1 Methodology Overview . . . 72
5.2 Adaptive Kernelized Maximum Minimum-Distance $(\mathrm{AKM^2D})$ Sensing . . 74
5.2.1 Formulation and Algorithm . . . 74
5.2.2 $\mathrm{AKM^2D}$ Sampling Properties . . . 75
5.2.3 Tuning Parameter Selection . . . 77
5.3 Mean and Anomaly Estimation Using Sparse Samples . . . 78
5.3.1 Robust Kernel Regression for Functional Mean Estimation . . . 78
5.3.2 Updating Probability $p_a(r_k)$ . . . 79
5.3.3 Sparse Kernel Regression for Clustered Anomaly Estimation . . . 80
5.4 Simulation Study . . . 81
5.5 Case Study . . . 85
5.6 Conclusion . . . 88

Chapter 6: Point Cloud Data Modeling and Analysis via Regularized Tensor Regression and Decomposition . . . 90

6.1 Basic Tensor Notation and Multilinear Algebra . . . 90
6.2 Tensor Regression Model with Scalar Input . . . 91

6.2.1 Regularized Tucker Decomposition 92

6.2.2 Customized Basis Selection 96

6.3 Simulation Study 97

6.4 Case Study 100

6.4.1 Handling unequal variances of residuals 102

6.4.2 Process optimization 104

6.5 Conclusion 105

Chapter 7: Conclusion 107

7.1 Summary of Original Contributions 107

7.2 Future Work 109

Appendix A: Appendix on "Anomaly Detection for Images and High-dimensional Spatial Functional Profile" 111

Appendix B: Appendix on "Anomaly Detection for Images and High-dimensional Spatial Functional Profile" 117

Appendix C: Appendix on "An Adaptive Framework for Online Sensing and Anomaly Detection " 124

Appendix D: Appendix on "Point Cloud Data Modeling and Analysis via Regularized Tensor Regression and Decomposition" 127

References 129

Vita 141

LIST OF TABLES

3.1 FPR, FNR, and computation time for line anomalies, clustered anomalies and scattered anomalies with $\delta = 0.3$ 35

3.2 The square root of mean square error of background and anamalies estimator with $\delta = 0.3$ 35

3.3 Case Study Sample Description 39

3.4 Computational time for all methods 40

4.1 Computation time of ST-SSD and other benchmark methods 61

4.2 Monitoring and diagnostics result when $\delta = 2$ and $\delta = 3$, (precision, recall and F, the larger the better; ARL, the smaller the better.) 62

5.1 Anomaly Detection Result with 250 and 400 sampled points 84

6.1 Cutting parameters for 9 experimental conditions 103

6.2 Gamma regression of $\|\hat{E}_i\|^2$ 104

A.1 FPR, FNR, and computation time for line , clustered and scattered anomalies with $\delta = 0.1, 0.2, 0.3$ 117

LIST OF FIGURES

1.1 A sample image for online surface inspection in rolling system 3

1.2 Examples of stress maps in photoelasticity experiment 3

1.3 Decomposition of image to background, defect, and noise 4

1.4 Example of HD streaming data with anomalies 6

1.5 Examples of cylindrical surface in 9 different settings 9

3.1 Simulated background from principal stress direction of a loaded circle . . . 31

3.2 The Simulated image of scattered and clustered anomalies 32

3.3 SSD decomposition results for scattered, line and clustered anomalies . . . 32

3.4 Detected anomalies for scattered, line, and clustered cases 34

3.5 Sensitivity study for line anomaly 36

3.6 Sensitivity study for scattered anomaly 37

3.7 Sensitivity study for clustered anomaly 37

3.8 Photo-elasticity experiment setup [102] 39

3.9 Detection results of Sample 1 and Sample 2 for all methods 41

4.1 Simulated images with both functional mean and anomalies at time $t = 201$ 58

4.2 Functional mean estimation results 60

4.3 Detection power comparison based on ARL 61

4.4 Detected anomalies by using different methods (incorrectly identified pixels are shown in red) 63

4.5 Detection results for rolling example at time $t = 97$ 65

4.6 Log of testing statistics in solar flare monitoring 67

4.7 Detection results in three solar frames at time $t = 192, 222, 258$ 68

4.8 Monitoring forging process using multi-channel tonnage signal 69

4.9 Tonnage Signal Diagnostics 70

5.1 Procedure of the proposed sampling algorithm 73

5.2 Behavior of $g(r)$ with the center point as anomaly point 77

5.3 Simulated images with both functional mean and anomalies 82

5.4 F-measure and Exploitation Ratio 84

5.5 Sampled point pattern for all methods for 250 and 400 points 85

5.6 Anomaly estimation result for all methods for 250 and 400 points 85

5.7 Guided wavefield experiment setup [14] 86

5.8 Energy map of the entire wavefield and detected anomaly 87

5.9 F-measure and Exploitation Ration 87

5.10 Sampled point pattern for all methods for 200 and 300 points 88

5.11 Anomaly estimation result for all methods for 200 and 300 points 88

6.1 Examples of generated point cloud for simulation study 100

6.2 SSE of the proposed methods with different magnitude in Case 1 and Case 2 101

6.3 Estimated and true coefficient for case 1 101

6.4 Estimated coefficient for case 2 102

6.5 Eigen-tensors with regularized Tucker decomposition 104

6.6 Result of tensor regression via regularized tucker decomposition 105

6.7 Simulated cylinder under the optimal setting (rotery speed: $80\mathrm{m}/\mathrm{min}$, cutting depth: $0.8250\mathrm{mm}$) 106

A.1 Anomalies detection comparison result for SSD and extended maxima transformation when $\delta = 0.3$ 116

SUMMARY

Analysis and modeling of the large-scale high-dimensional data is very important. This thesis focuses on modeling the high-dimensional data obtained from sensors for assessment of system performance, early detection of system anomalies, intelligent sampling and sensing for data collection and decision making to achieve optimal system performance. The developed methodology should be efficient and scalable and can be applied for data with complex heterogeneous data structure to extract information or useful features.

The research topic that Chapter 3 focuses on is to detect anomalies from high-dimensional functional data. We first study the problem of detecting anomaly in high-dimensional spatial profile in Chapter 3. In various manufacturing applications such as steel, composites, and textile production, anomaly detection in noisy images is of special importance. Although there are several methods for image denoising and anomaly detection, most of these perform denoising and detection sequentially, which affects detection accuracy and efficiency. Additionally, the low computational speed of some of these methods is a limitation for real-time inspection. In Chapter 3, we develop a novel methodology for anomaly detection in noisy images with smooth backgrounds. The proposed method, named smooth-sparse decomposition, exploits regularized high-dimensional regression to decompose an image and separate anomalous regions by solving a large-scale optimization problem. To enable the proposed method for real-time implementation, a fast algorithm for solving the optimization model is proposed. Using simulations and a case study, we evaluate the performance of the proposed method and compare it with existing methods. Numerical results demonstrate the superiority of the proposed method in terms of the detection accuracy as well as computation time.

In Chapter 4, we extend this to spatial-temporal functional data. High dimensional data monitoring and diagnosis has recently attracted increasing attention among researchers as well as practitioners. However, existing process monitoring methods fail to fully utilize the

information of high dimensional data streams due to their complex characteristics including the large dimensionality, spatio-temporal correlation structure, and non-stationarity. In Chapter 4, we propose a novel process monitoring methodology for high-dimensional data streams including profiles and images that can effectively address foregoing challenges. We introduce spatio-temporal smooth sparse decomposition (ST-SSD), which serves as a dimension reduction and denoising technique by decomposing the original tensor into the functional mean, sparse anomalies, and random noises. ST-SSD is followed by a sequential likelihood ratio test on extracted anomalies for process monitoring. To enable real-time implementation of the proposed methodology, recursive estimation procedures for ST-SSD are developed. ST-SSD also provides useful diagnostics information about the location of change in the functional mean. The proposed methodology is validated through various simulations and real case studies.

In Chapter 5, we focus on the adaptive sampling for high-dimensional functional data. In point-based sensing systems such as coordinate measuring machines (CMM) and laser ultrasonics where complete sensing is impractical due to the high sensing time and cost, adaptive sensing through a systematic exploration is vital for online inspection and anomaly detection. Most of existing sequential sampling methodologies focus on reducing the overall fitting error for the entire sampling space. However, in many anomaly detection applications, the main goal is to accurately detect and estimate sparse anomalous regions. In Chapter 5, we develop a novel framework named Adaptive Kernelized Maximum-Minimum Distance $(\mathrm{AKM^2D})$ to speed up the inspection and anomaly detection process through an intelligent sequential sampling scheme integrated with fast estimation and detection. The proposed method balances the sampling efforts between the space filling sampling (exploration) and focused sampling near the anomalous region (exploitation). The proposed methodology is validated by conducting simulations and a case study of anomaly detection in composite sheets using a guided wave test.

Chapter 6 explores the penalized tensor regression to model the tensor response data

with the process variables. This method is inspired by the advance 3D metrology technologies such as Coordinate Measuring Machine (CMM) or laser 3D scanners. These techniques has facilitated the collection of massive point cloud data, beneficial for process monitoring, control and optimization. However, due to their high dimensionality and structure complexity, modeling and analysis of point clouds is a challenge. In Chapter 6, we represent point clouds using tensors and propose regularized tucker decomposition and regularized tensor regression to model the variational patterns of point clouds and link them to process variables. The performance of the proposed method is evaluated through simulation and a real case study of turning process optimization.

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Research Background

Nowadays most manufacturing processes are instrumented with sensing systems comprised of hundreds of sensors to monitor process performance and product quality. The low implementation cost, high acquisition rate, and high variety of such sensing systems lead to rich data streams that provide distinctive opportunities for performance improvement. Real-time process monitoring and control, accurate fault diagnosis, and online product inspection are among the benefits that can be gained from effective modeling and analysis of streaming data. However, the complex characteristics of these data streams pose significant analytical challenges yet to be addressed. Common characteristics of these data streams include 1) *High variety*: Various types of sensors generate a high variety of data streams, including profiles or waveform signals (e.g. an exerted force profile during a forging operation [1]), images (e.g. an image of a bar surface after rolling [2]), and videos (e.g. a video of a industrial flame in steel tube manufacturing [3]); 2) *High dimensionality*: A typical image used for surface inspection is on the order of 1M pixels [2]; 3) *High velocity*: In recent years, the speed of data collection has significantly increased so that it can keep up with almost any production rate. For example, a commercially available ultrasonic sensor can easily record data at the rate of 1KHz, and a high-speed industrial camera is capable of scanning a product surface with the rate of 80 million pixels per second or faster [2]; 4) *Spatial and temporal structure*: Another layer of complexity arises because of the spatio-temporal structure of streaming data. Data points in a profile or pixels within an image are spatially correlated (e.g. neighbor pixels often exhibit high correlations) and corresponding data points or pixels across sequential samples are often temporally correlated with

non-stationary behavior.

1.2 Specific Research Topics

1.2.1 Anomaly Detection for High-dimensional Functional Profiles

Image is one of the most popular examlpe of the high-dimensional functional data. Image sensing systems have been widely deployed in a variety of manufacturing processes for online process monitoring and fault diagnosis. The reasons for this range from their low implementation cost and high acquisition rate of image sensors to the rich process information they provide. One of the main applications of these systems is real-time product inspection in which a snapshot of a product or part is analyzed to detect defects or anomalies. One example is in continuous casting manufacturing where molten metal is solidified into a semi-finished billet used in the subsequent rolling process. To inspect the quality of billets and detect anomalies on their surfaces, a vision sensing system is set up to take snapshots of billets at short time intervals. A sample of the surface image with a vertical curved line defect is shown in Figure 1.1. Considering the high speed of production, an automatic, quick, and accurate image analysis technique is crucial to an effective quality inspection and anomaly detection system.

Another example of image-based quality inspection, as shown in Figure 1.2, is in the photoelasticity test [4]. The photoelasticity test is a non-destructive evaluation method used for stress and strain analysis of translucent parts or material. The output is often presented by a colormap in which regions with high-tensile stress, often associated with anomalies, are shown by warm colors. The stress maps of a silicon wafer sample and a composite laminate with surface indentation are shown in Figure 1.2a and Figure 1.2b, respectively. Other applications where image-based inspection and anomaly detection have been used include the rolling process [2], composite material fabrication [5], liquid crystal display manufacturing [6], fabric and textile manufacturing [7], and structural health monitoring [8], to name a few. Although anomalous regions are normally clear for human inspection,

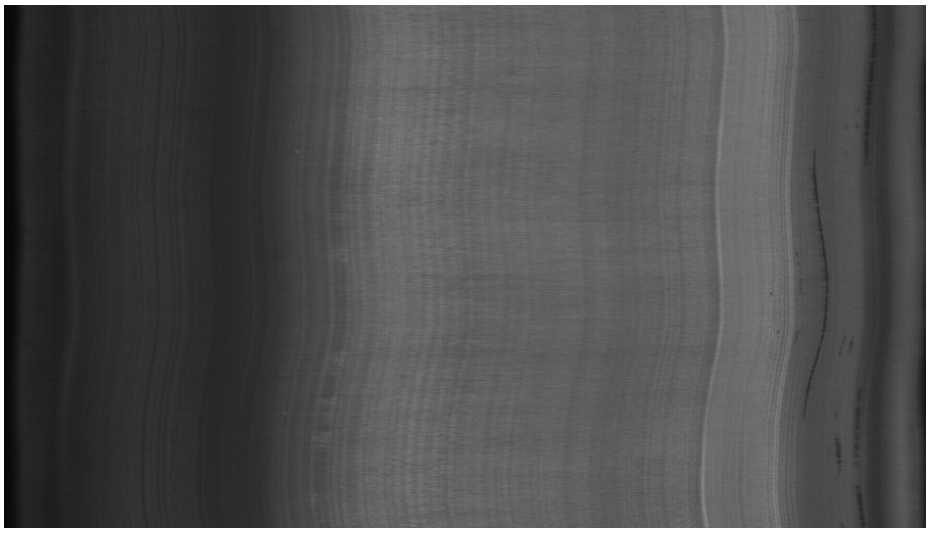
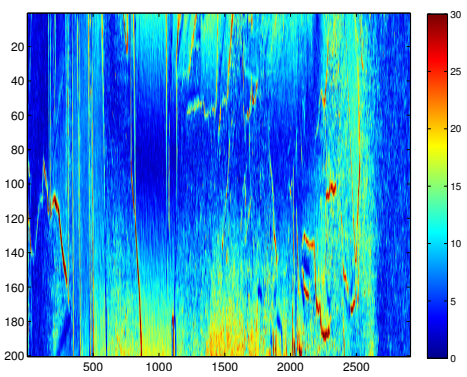
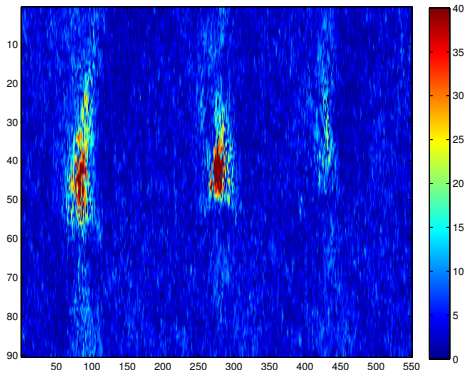


Figure 1.1: A sample image for online surface inspection in rolling system



(a) Stress map of multi-crystalline bending beam



(b) Stress map of composite with surface indentation

Figure 1.2: Examples of stress maps in photoelasticity experiment

developing an automatic algorithm that can accurately detect and separate these regions from the image background in real time is imperative for effective process monitoring.

In Chapter 3, we propose an accurate and fast method for image-based anomaly detection that overcomes the drawbacks of existing two-step methods. Our method integrates smoothing and anomaly detection tasks into one step through a novel smooth-sparse decomposition (SSD) approach. SSD decomposes an image into three components: namely, the smooth image background, the sparse anomalous regions, and the random noises, as illustrated in Figure 1.3. In addition to anomaly detection, SSD helps retrieve an image

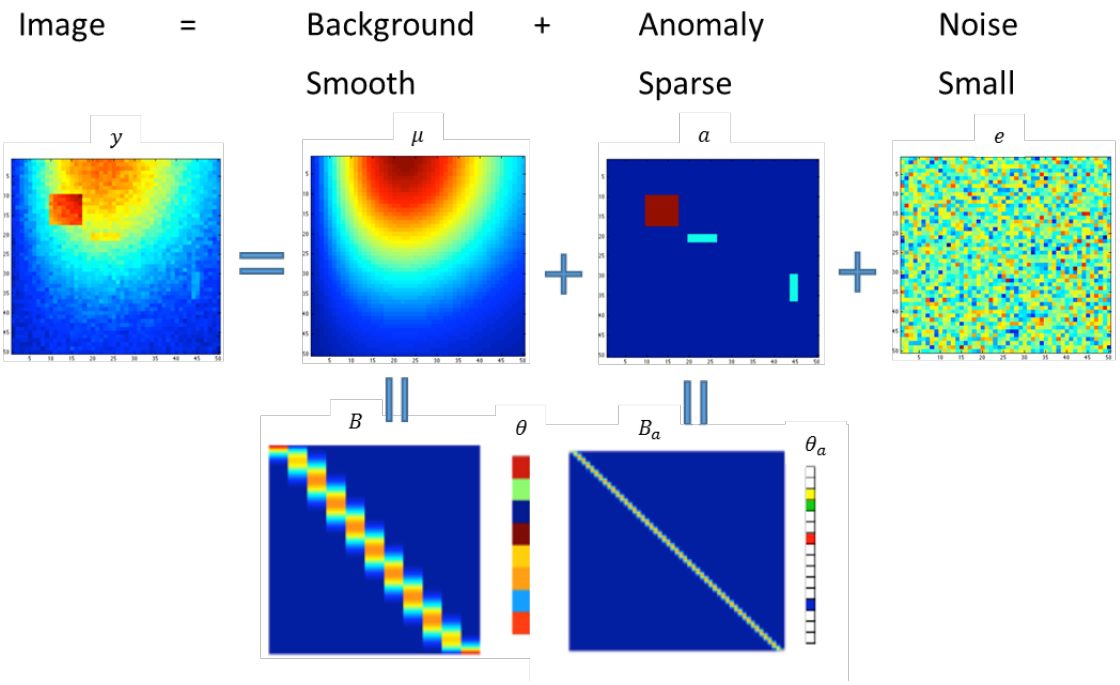


Figure 1.3: Decomposition of image to background, defect, and noise

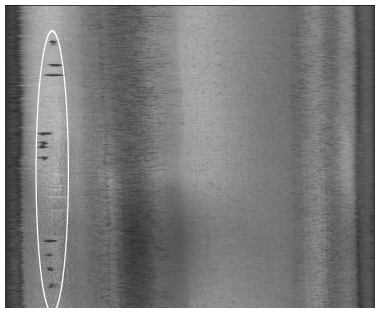
by removing anomalies and random noises. SSD is developed based on the premise that the image background is smooth and anomalous regions are sparse or can be expressed by sparse bases. Decomposition is achieved by constructing a penalized non-parametric regression model that enforces background smoothness and anomaly sparsity through penalty terms added to the loss function. In order to estimate the parameters of the regression model and perform the decomposition in real-time, we propose efficient optimization algorithms.

1.2.2 Anomaly Detection for High-Dimensional (HD) Functional Data Streams with Smooth Spatial-temporal Structures

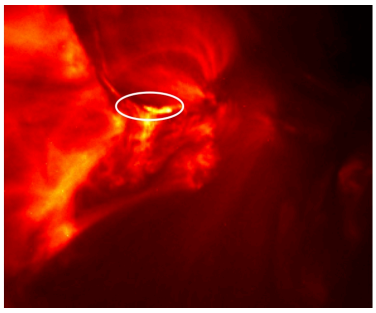
We then extend the anomaly detection for HD functional/spatial data to HD spatial-temporal data streams. Examples of such high dimensional (HD) data are shown in Figure 1.4. In Figure 1.4a, a sample of a bar surface used for monitoring of a rolling process is shown [2]. In the second example, shown in Figure 1.4b, a sequence of solar images captured by a satellite is used to monitor solar activities and detect solar flares. Figure 1.4c shows a

sample of normal and faulty multi-channel tonnage profiles used for monitoring a forging process [1]. As can be seen from the figures and clips, an in-control HD data stream can typically be represented by a functional mean with a smooth spatial structure that gradually changes over time. However, this gradual change manifests inherent dynamics of the process and should not be considered as an out-of-control situation. Anomalies, on the other hand, are in the form of abrupt changes with a spatio-temporal structure different from the functional mean. The smooth temporal change of the functional mean may significantly increase the false alarm rate of a monitoring procedure if not appropriately modeled. This makes monitoring of HD data streams even more challenging. Most of existing HD monitoring methods fail to model the temporal trend of the functional mean, and only focus on change detection by assuming that the in-control functional mean is constant over time.

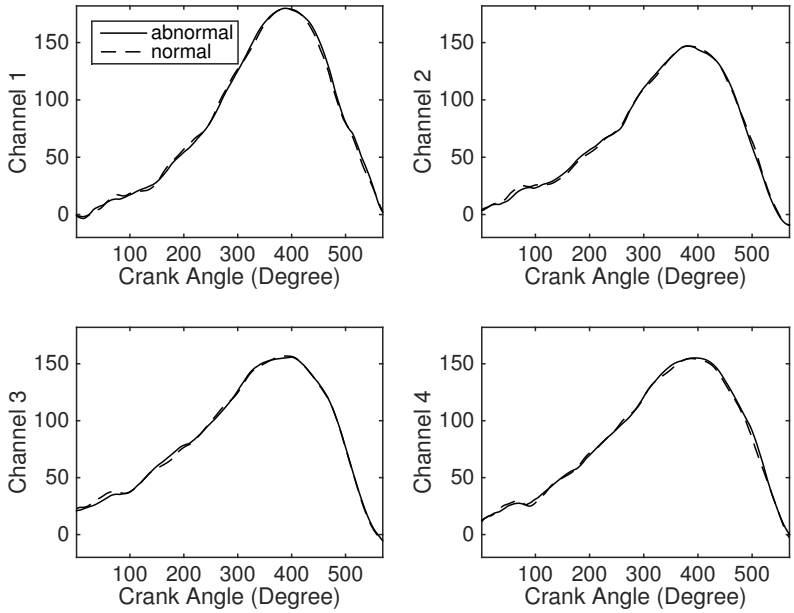
The Chapter 4 develops a new scalable spatio-temporal methodology for real-time monitoring and diagnosis of HD high-velocity streaming functional data with time-varying means. This methodology is also capable of identifying the location of the change, which is important for diagnosis. Our proposed methodology is inspired by the recent development of smooth-sparse decomposition (SSD) for anomaly detection in images [9]. SSD can separate anomalies from the image background by utilizing the spatial structure of an image. The key idea is to extend the SSD methodology so that it can incorporate temporal information of an HD data stream in addition to the spatial information of a single sample. However, this extension is nontrivial because adding the time dimension significantly increases the dimensionality of the problem, given the high rate data acquisition. In Chapeter 3.2, we begin with extending the SSD method to spatio-temporal SSD so it can include temporal information and model smooth temporal trend of a data stream. Assuming that the functional mean of the data stream is spatially and temporally smooth and process changes/anomalies are non-smooth and sparse in a certain basis representation, our proposed spatio-temporal SSD decomposes an HD data stream into a smooth spatio-temporal functional mean, sparse anomalous features and random noises. This model serves as a



(a) Surface image of a steel bar



(b) Temperature map of sun surface



(c) Tonnage signals in a forging process

Figure 1.4: Example of HD streaming data with anomalies

dimension reduction technique, which reduces the HD data stream to a small set of features. We then develop recursive estimation procedures that significantly reduce the computational complexity and enable the real-time implementation of the method. Finally, we combine the proposed model with a likelihood-ratio test (LRT) to monitor the process based on the detected anomalies/features.

1.2.3 An Adaptive Framework for Online Sensing and Anomaly Detection

In metrology and non-destructive evaluation (NDE), various point-based sensing systems are used for quality inspection and anomaly detection. Examples include, touch-probe coordinate measuring machines (CMM) used for measuring the dimension accuracy [10], and non-destructive methods such as guided wave-field tests (GWT) [11] and laser ultrasonics [12], utilized for defect detection and quantification in composite sheets. Most point-based sensing systems are only capable of measuring one point at a time, resulting in a time-consuming procedure not scalable to online inspection of large areas. For example, using a touch-probe CMM, it may take more than eight hours to measure one typical batch of wafers that includes 400 wafers of 11" diameters [13]. Also, using GWT, the high-resolution inspection of a composite laminate of size 1m^2 may take up to four hours [14]. However, due to the fact that anomalies are often clustered and sparse, one can use a sequential and adaptive sampling strategy to reduce the measurement time by reducing the number of sampled points. Therefore, the objective of the Chapter 5 is to propose a new adaptive sensing framework along with estimation procedures for online anomaly detection. The immediate benefit of the proposed framework is to help scale up point-based sensing methods so that they can be used for in-situ inspection. An effective adaptive sensing strategy should consist of two major elements: First, it should randomly search the entire space (exploration) to spot anomalous regions and recover the functional mean; and second it should perform focused sampling on areas near the anomalous region (exploitation) to determine the size and the shape of anomalies. To achieve this, the following two challenges should be addressed: 1) how to intelligently decide on the location of the next sampled point; and 2) how to estimate anomalous regions as well as the functional mean online based on the sparsely sampled points. In Chapter 5, we will address the first challenge by proposing a new sensing strategy named Adaptive Kernelized Maximum Minimum-Distance (AKM^2D) combining the computer design of experiment approach for the random exploration of the entire space and the Hilbert Kernel approach

[15] for the focused sampling in anomalous regions (exploitation). To address the second challenge, we propose a modeling framework based on robust kernel regression for estimating the background (profile mean) and sparse kernel regression for estimating and separating anomalies. In order to perform both estimation and adaptive measurement in real-time, we also propose efficient optimization algorithms.

1.2.4 Modeling the High-dimensional Structured Point Cloud with Process Variables

Nowadays, a variety of products with complex shape and geometry can be manufactured due to the advancement of manufacturing technologies such as addictive manufacturing [16]. However, due to the variability of the manufacturing process, the actual dimensions and geometry may deviate from its ideal or nominal shape [17]. Therefore, finding the relationship of the 3D geometry of produced parts with the process variables and machine settings is vital for process modeling and optimization. Modern sensing technology enables fast and accurate measurement of parts geometry and dimensions [18]. For example, touch-probe Coordinate Measuring Machines (CMM) is capable of measuring the 3D geometry of parts within only a few minutes [10]; recent optical systems like laser scanners are capable of measuring complex geometries with high-sampling frequency (ranging from 10 to 500 KHz) within seconds [19]. Most 3D metrology technologies share a similar measurement mechanism in which the actual coordinates of points-of-interest on the surface of an object is measured and stored [20]. This mechanism results in a set of point in a 3D space, referred to as a point cloud [21]. In most 3D metrology systems, the points-of-interests are defined on a pre-specified grid. In this chapter, we refer to this type of data as structured point-cloud, commonly found in dimensional metrology [22].

The objective of Chapter 6 is to develop a tensor regression framework to model a tensor response as a function of some scalar predictors. To achieve this, we propose to represent the response tensor on a set of basis which helps significantly reduce the dimensionality and consequently facilities the parameter estimation. For the basis selection, we will intro-

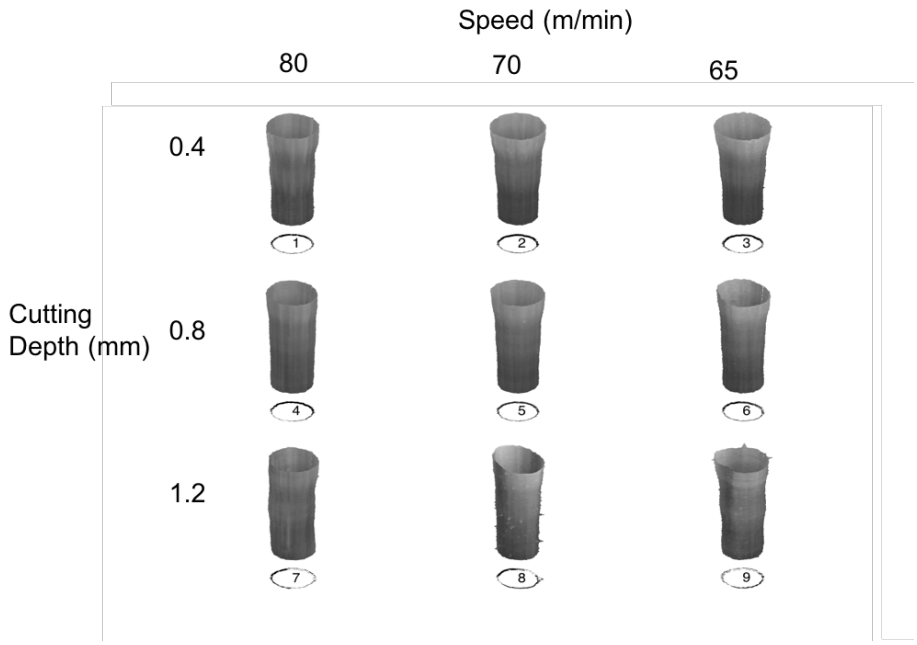


Figure 1.5: Examples of cylindrical surface in 9 different settings

duce two frameworks, namely the Regularized Tucker Decomposition Regression (RTDR) and Regularized Tensor Regression (RTR). For RTDR, we will learn the basis from the data directly by Regularized Tucker Decomposition. It is worth noting that the proposed Regularized Tucker Decomposition can extract useful variational patterns of the tensor response, which is useful not only for regression analysis but also in other applications such as fault identification and diagnosis. For RTR, we project the response tensor on a set of pre-defined basis such as Splines with roughness penalization to control the smoothness of the estimated tensor response.

1.3 Thesis Organization

Thesis is organized as follows: Chapter 2 reviews existing literature pertaining to several research areas: image based anomaly detection, monitoring of HD functional data stream, adaptive sampling strategy for HD data, and HD structured point cloud modeling. Chapter 3 introduces the proposed one-step framework that can identify anomaly from spatial HD functional data Section 4 extends this framework to dynamic HD functional data stream.

Chapter 5 describes the proposed adaptive sampling strategy that balances exploration of the entire sampling space and focus sampling near the anomalous regions. Chapter 6 introduces the proposed regularized tensor regression to model the HD structured point cloud with process variables. Chapter 7 summarizes the thesis and introduces future research directions.

CHAPTER 2
LITERATURE REVIEW

The high dimensionality and complex spatial structure of images coupled with measurement noises that reduce contrast between anomalies and the background pose significant challenges in developing real-time anomaly detection methods. Owing to its importance, anomaly detection in images has been extensively studied in the literature, and considerable research has been conducted to address these challenges. Most of existing image-based anomaly detection methods follow a common two-step procedure in which first, a smoothing or denoising method such as spline [23] or wavelet analysis [24] is applied to reduce image noises, then a detection algorithm is exploited to identify anomalous regions. The major problem of the two-step approach is that the smoothing step often blurs the sharp boundaries of anomalous regions, which in turn makes the detection step challenging [25]. [26] and [27] developed edge-preserved smoothing techniques to preserve the boundary structure applying the smoothing algorithm. Although effective methods for denoising of images with a general background structure, they should be followed by an anomaly detection or image segmentation algorithm to identify anomalous regions. This may increase the analysis time.

Related to anomaly detection in single image or functional profile, considerable research has been conducted on anomaly detection in single image observation, most of which is based on a two-step approach comprised of denoising and detection. For the denoising step, smoothing methods such as spline regression [23], B-Spline [28], Penalized Spline [29], kernel smoothing [30], and various filtering techniques [31] have been widely used. To preserve the boundaries of anomalous regions while applying smoothing, [26] and [27] developed edge-preserved smoothing techniques by assigning small weights to the observations located on either side of the boundary. In the area of detection, a considerable

body of research focuses on anomaly detection in images with patterned background - for examples, images taken from textured materials. Most methods in this category rely upon identifying the areas that differ from the background pattern. Examples include analysis of variance [6], the filter technique [32], SVD decomposition [33], wavelet transformation [5], Fourier transformation [34], etc. These methods, however, are not effective when the background of an image is smooth.

[35] categorized current anomaly detection methods, applicable to images with smooth backgrounds, into three categories: edged-based, thresholding-based, region-based methods. Edge-based methods focus on locating sharp discontinuities in the image. Most traditional edge detection methods are based on gradient estimation, i.e., the estimation of first or second order derivatives of the image intensity function. For example, Sobel [36] and Prewitt [37] are the two widely used first-order derivative operators for edge detection. Laplacian operator or Laplacian of Gaussianedge detector [38] are second-order derivative operators for edge detection. However, gradient-based edge detection methods are not robust to noise because the edge detection mask is usually small. To overcome this problem, [39] proposed jump regression that utilizes local surface information. Jump regression enables precise estimation of jump locations or edges on a smooth background in a noisy setting [39, 26]. The output of edge detection algorithms are often discontinuous pixels, which may not form a closed-boundary region or continuous curves. Consequently, most edge-based methods require a post-processing step such as edge linking or filling algorithm to link them for creating closed-boundary regions and fill the area inside these regions. The main problem of edge-based methods is that even after applying edge-linking algorithms, the detected edges still do not form a closed region [35]. To address this, [40] developed a curve estimation algorithm to effectively link scattered pixels with a closed curve. [41] also developed a local non-parametric segmentation methods for spotted microarray images. [42] developed a multi-stage semi-automated procedure to extract the shape information of nanoparticles based on the boundary points from edge detectors. However, these

methods are case-specific (i.e. for spotted microarray image detection and nanoparticles detection) and may not work well for other types of anomalies such as scattered or line defects. Moreover, they require information of the centroid location, which, in some cases, may be difficult to find or estimate. Computational speed is another issue because these methods often entail multiple processing steps.

Unlike edge-based methods, thresholding-based methods utilize the intensity difference between anomalies and the background to find closed-boundary regions directly. The thresholding algorithm can be further classified into two categories: global thresholding and local thresholding. Otsu's method [43] is one of the global thresholding methods that uses a single value obtained by minimizing in-class variation to globally threshold the entire image and produce a binary map for anomalous regions. However, as pointed by [44], global thresholding algorithms do not perform well when large intensity variation exists in the image background. On the contrary, local thresholding methods (e.g., [45], and [46]) divide an image into small regions and then perform the thresholding locally in each region. There are two main problems with local thresholding methods: First, in non-defective regions, local thresholding algorithms still pick a threshold value, which leads to false detection. Second, anomalous regions are often larger than the thresholding window, which leads to inconsistent thresholding of the anomalous region.

Similar like local thresholding methods, region-based methods give closed-boundary regions. For example, the "seeded region growing" method [47] starts from a small initial region (pixels) and grows it by exploring neighbor pixels and adding pixels similar to the initial ones. However, region-based algorithms often require a small set of initial anomalous pixels from which the region can be grown. Consequently, when lacking the location information of anomalies or in the case of multiple anomalous regions, this method is not practical. Extended-maxima transformation [48] is one of the morphological filtering methods that can be used for anomaly detection . This method searches for connected pixels with a constant intensity that are distinguished from the background by an exter-

nal boundarywith a different intensity. However, this method does not take advantage of the smooth structure of the background and anomalies, thus may fail when the intensity contrast between anomalies and the background is small.

2.1 Monitoring and Diagnosis of High-dimensional Streaming Functional Data

There is a considerable body of literature on monitoring and diagnosis of HD streaming data. Current research in this area can be classified into three groups: monitoring methods for HD multivariate data streams, profile monitoring techniques, and monitoring methods based on dimension reduction. In the first group, HD data are treated as multiple univariate data streams. For example a profile stream with a length of 200 generates 200 individual data streams. Under the assumption that data streams are independent, [49] proposed a monitoring scheme based on the sum of the local CUSUM statistics for individual streams. [50] extended this method and developed an adaptive sensing scheme assuming that only partial observations are available. [51] developed a powerful goodness-of-fit test for monitoring independent HD data streams. However, these methods assume that the data streams are independent and therefore, ignore their temporal and spatial structures. To monitor univariate data streams with a temporal trend, [52] and [53] combined nonparametric regression with longitudinal modeling techniques. However, they did not consider the spatial structure of the functional mean and anomalies. The literature on nonlinear profile monitoring is rich, which includes various parametric and nonparametric methods. For example, for monitoring smooth profiles, there are various nonparametric methods based on local kernel regression [54, 55] and splines [56]. [57] used wavelets to model and monitor non-smooth profiles. These methods, however, are not applicable to profiles with time-varying means. Moreover, most of these methods are specifically designed for profile motioning, and their generalization to image and video streams is nontrivial. Among the dimension reduction approaches, principal component analysis (PCA) is the most popular method for HD data monitoring because of its simplicity, scalability, and data compression capability.

For example, [58] used PCA to reduce the dimensionality of streaming data and constructed T^2 and Q charts to monitor extracted features and residuals, respectively. [59] proposed a monitoring approach for multichannel signals by combining multivariate functional PCA and change-point models. [3] developed a tensor-based principal component analysis that can model both the spatial and spectral structures of an image sequence. [60] proposed a multi-resolution PCA for profile monitoring by integrating PCA with wavelets. The main drawback of PCA-based methods is that they cannot be directly used for non-stationary data streams with a time-varying mean. To address the drawbacks of existing methods, we propose a new spatio-temporal smooth sparse decomposition for monitoring and diagnosis of HD data streams.

2.2 Adaptive Sampling and Sensing for HD spatial profiles

Existing adaptive sampling/sensing strategies in the literature can be classified into three groups: the multi-resolution grid strategy, sequential design of experiments (SDOE), and representative points selection. The multi-resolution grid sensing has been widely used in practice. It begins with sensing over a coarse (low-resolution) grid to estimate the underlying functional mean (e.g. the image background in 2D measurements) and find the rough locations of anomalies. Then, sensing is continued over a finer (high-resolution) grids around the identified anomalies to estimate the anomaly shape and size. The performance of this method depends on the predefined size of fine grids, which should be specified based on the size and shape of anomalies. Since such information may not be available in advance, this method may result in either over-sampling or poor anomaly detection caused by under-sampling. In the SDOE class, [61] classified current sequential design of experiment methods into model-based and distance-based (space-filling) depending on the criterion defined for the sequential selection of the sampled points. Model-based methods include maximizing the expected improvement criterion [62, 63], minimizing the prediction error, minimizing the variance of the parameter estimates, e.g., D-optimal design

[64], and optimizing a composite index [13]. Among the distance-based models, sequential LHD design [65, 66] and Sequential maximin design [67, 68] are widely used. However, the main problem of SDOE methods is that they only focus on improving the estimation of the functional mean over the entire sampling space without considering potential anomalies and non-smooth features. In the third group, [69] proposed the minimum energy design that selects representative points based on a known distribution over the design space and sequentially chooses the next design points based on a criterion minimizing the total potential energy. However, the main problem of applying this approach for online anomaly detection is that the anomaly distribution is often unknown a priori. Therefore, it lacks the ability of focused sampling near anomalous regions.

Another relevant body of literature deals with function estimation in the presence of anomalies. Robust kernel regression [70] and robust spline estimation [71] are among these methods. However, their main focus is the estimation of the functional mean not the anomaly, and hence, they do not fully consider the spatial structure of anomalies. To address this issue, [72] proposed smooth-sparse decomposition (SSD) for anomaly detection in temporal and/or spatial profiles. SSD can separate anomalies from the functional mean by utilizing the spatial structure of both the functional mean and anomalies. SSD, however, can only work efficiently when measurements are dense, hence, not applicable in point-based sensing and inspection systems.

2.3 Modeling Point Cloud data

High dimensionality and complex structure of point clouds pose significant challenges in data analysis. In the literature, there exist a variety of methods for point cloud representation and surface reconstruction. In many applications, point clouds are converted to polygon or triangle mesh models [73], NURBS surface models [74], or CAD models through surface reconstruction techniques such as Delaunay triangulation [75], alpha shapes [76], and ball pivoting [77]. Although these techniques are effective in providing a compact

representation of point clouds, they are not capable of modeling the relationship between point clouds and some independent variables, which is important in many applications. In this area, the literature on modeling and analysis of point clouds can be classified into two general categories depending on the objectives: *(i) process monitoring* and *(ii) process modeling and optimization*.

Research in the first category mainly focuses on finding unusual patterns in the point cloud to detect out-of-control states and the corresponding assignable causes. For example, [78] combined parametric regression with univariate and multivariate control charts to quantify three dimensional surfaces with spatially correlated noises. However, this model assumes that a parametric model exists for 3D point clouds, which may not be available for surfaces with complex shapes. To address this challenge, [79] proposed to use QQ plots to transform the high-dimensional point cloud monitoring problem into a linear profile monitoring problem. However, due to the use of Q-Q plot, this approach fails to capture the spatial information of the 3D point cloud. [80] applied Gaussian Process to model and monitor 3D surfaces with spatial correlation. However, Gaussian Process can be inefficient for high-dimensional point clouds such as those in our application.

The main objective of the second category is to build a response model of a structured point cloud as a function of some controllable factors, and then use this model to find the optimal control settings to minimize the dimensional and geometric deviations of produced parts from its nominal values. Point clouds are often represented by point locations in the Cartesian coordinate system. However, a structured point cloud can be represented compactly in a multidimensional array (tensor) because of the grid structure. Therefore, modeling the structured point cloud with some controllable factors can be considered as a tensor regression problem.

2.4 Tensor-on-scalar regression

While many literature exists on regression with high dimensional functional data, there has been little work focusing on functional regression with a tensor response. We would review three major lines of research in this area. The first line of research focuses on regression models with multivariate vector response. For example, one can use Principal Component Analysis (PCA) [81] to reduce the dimensionality of the response and then build a regression model for the estimated PC scores. [82] proposed a functional-on-scalar regression, in which the functional response is linked with scalar predictors via a set of functional coefficients to be estimated in a predefined functional space with a certain penalty function. Other methods such as partial least squares [83] or sparse regression [84] are also capable of regressing a multivariate or functional response on scalar predictors. Although these methods are effective for modeling high-dimensional vectors, they are inadequate for analysis of structured point clouds due to the ultrahigh dimensionality as well as their complex tensor structure [85]. The second line of research focuses on regressing a scalar response with the tensor covariates. For example, [85] proposed a regression framework in which the dimensionality of the tensor covariates is substantially reduced by applying low-rank tensor decomposition technique leading to efficient estimation and prediction. The third line of research, directly related to our problem, is on modeling tensor response with scalar or vector predictors. For example, one popular method is to regress each entry of the response tensor independently on scalar predictors [86] and generate a statistical parametric map of the coefficients across the entire response tensor. A smoothing approach is often required as a preprocessing step to remove the noise in the tensor reponse. For example, [87] proposed a multi-scale adaptive approach to smooth the tensor response first before the regression model. However, the major drawback of this approach is that all response variables are treated independently and important spatial correlation is often ignored. To encounter this problem, [88] built a parsimonious linear tensor regression model by assuming that only

part of tensor response depends on the scalar predictors. Although this sparsity assumption is valid in Neuroimaging applications, it may not be valid in other point cloud applications such as the one discussed in Chapter 6.

CHAPTER 3
ANOMALY DETECTION FOR IMAGES AND HIGH-DIMENSIONAL SPATIAL FUNCTIONAL PROFILE

In this chapter, we develop a novel methodology for anomaly detection in noisy images with smooth backgrounds. The proposed method, named smooth-sparse decomposition, exploits regularized high-dimensional regression to decompose an image and separate anomalous regions by solving a large-scale optimization problem. To enable the proposed method for real-time implementation, a fast algorithm for solving the optimization model is proposed.

The reminder of this chapter is organized as follows. Section 3.1 elaborates the one-step SSD approach for anomaly detection and presents efficient optimization algorithms for the real-time implementation of SSD. In Section 3.2, we use simulated image data with different anomaly structures to evaluate the performance of the proposed SSD method and compare it with some existing two-step methods in terms of detection accuracy as well as computation time. In Section 3.3, we illustrate a case study in which we apply the proposed SSD for anomaly detection in composite laminates and silicon wafers. We conclude the chapter with a short discussion in Section 3.4.

3.1 Smooth-Sparse Decomposition (SSD)

In this section, we present the penalized nonparametric regression model used for SSD and propose efficient algorithms for its implementation. For simplicity, we first discuss the methodology for one-dimensional (1-D) signals and then generalize it to n-D images ($n > 1$). When the underlying manufacturing process is unstable, faulty parts may be produced. In this case, the measured signal is comprised of not only the functional mean and noises but also anomalies as shown in Figure 1.2a and Figure 1.2b. In this chapter,

anomalies are defined as faults whose functional structure differ from the functional mean of the background and their magnitude is larger than that of the noise. Therefore, SSD aims to decompose the signal into a smooth functional mean, sparse anomalous regions, and random noises. SSD aims to decompose the signal into a smooth functional mean, sparse anomalous regions, and random noises. Specifically, the signal is decomposed as $y = \mu + a + e$, where μ is the smooth mean of the signal, a is the vector of anomalies assumed to be sparse in a certain functional space, and e is the vector of random noises. We further expand the mean and anomalies using a smooth basis (e.g., spline basis) denoted by B and B_a, respectively. Consequently, the signal decomposition model can be rewritten as $y = B\theta + B_a\theta_a + e$, where θ and θ_a are, respectively, the basis coefficients corresponding to μ and a. Least square regression is used to estimate the model parameters (i.e., θ and θ_a). To ensure the smoothness of the estimated mean and the sparsity of the detected anomalies, the least square loss function is augmented by L_1 and L_2 penalties, which results in the following penalized regression criterion:

$$\underset{\theta,\theta_a}{argmin} \|e\|^2 + \lambda\theta^T R\theta + \gamma\|\theta_a\|_1, \text{ subject to. } y = B\theta + B_a\theta_a + e \tag{3.1}$$

where $\|\cdot\|$ and $\|\cdot\|_1$ are L_2 and L_1 norm operators, and λ and γ are tuning parameters to be determined by the user. R is the roughness matrix, and can be defined as $R = D^T D$ in the 1D case, which is related to the difference between the nearby spline coefficients, i.e., $\|\Delta^d\theta\|^2$, where Δ^d is the d^{th} order difference operator. It is not hard to show this penalization term can also be written as $\|\Delta^d\theta\|^2 = \theta^T R\theta$, in which $R = D^T D$, D is the d^{th} order difference matrix. For example, if $k = 1$, $D = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix}$. Therefore, the L_2 penalty term, $\lambda\theta' R\theta$, regularizes the level of smoothness of the mean function, while the L_1 penalty term, $\gamma\|\theta_a\|_1$, encourages the sparsity of the anomalous regions. The constraint guarantees that the signal can be reconstructed using the linear combination of

the estimated components. Note that if $\lambda = 0$, the SSD model boils down to LASSO [89].

3.1.1 Optimization Algorithms for SSD

The loss function in (A.1) is convex and can be solved via general convex optimization solvers like the interior point method [90]. However, the interior point method is often slow for large-scale problems and hence cannot be used for real-time inspection and monitoring. In this section, we propose a set of efficient algorithms for the real-time implementation of SSD for two cases: orthogonal- and general-anomaly basis (i.e., B_a).

Orthogonal Basis B_a

If the basis B_a is orthogonal, the block coordinate descent (BCD) method is used to break down the SSD model into two simpler optimization problems. The BCD is a class of algorithms that groups domain variables into different blocks and finds a local minimum for a function by iteratively minimizing this function with respect to one block given all other blocks. By defining θ and θ_a as two variable blocks, a two-step iterative algorithm based on the BCD can be used to find the minimizer of (A.1). In each iteration k, given $\theta_a^{(k-1)}$, the SSD loss function in (A.1) reduces to a weighted ridge regression, which has a closed-form solution in the form of $\theta^{(k)} = (B^T B + \lambda R)^{-1} B^T (y - B_a \theta_a^{(k-1)})$. Equivalently, it can be shown that $y^{(k)} = H(y - B_a \theta_a^{(k-1)})$, where $H = B(B^T B + \lambda R)^{-1} B^T$. In the second step of the optimization algorithm, according to Proposition 1, given $\theta^{(k)}$ or $\mu^{(k)}$, $\theta_a^{(k)}$ is updated by a simple soft-thresholding operation.

Proposition 1. *If B_a is orthogonal, in iteration k, the subproblem $\theta_a^{(k)} = \operatorname{argmin}_{\theta_a} \|y - B\theta^{(k)} - B_a\theta_a\|^2 + \gamma\|\theta_a\|_1$ has a closed-form solution in the form of $\theta_a^{(k)} = S_{\frac{\gamma}{2}}(B_a^T(y - B\theta^{(k)}))$, in which $S_\gamma(x) = \operatorname{sgn}(x)(|x| - \gamma)_+$ is the soft-thresholding operator, and $\operatorname{sgn}(x)$ is the sign function and $x_+ = \max(x, 0)$.*

The proof is given in Appendix A.

Algorithm 1: Optimization algorithm for SSD based on BCD method for orthogonal B_a

initialize
  Choose a basis for the background as B and for the anomalous regions as B_a
  $\theta_a^{(0)} = 0$, $k = 1$
end
while $\left\|\theta_a^{(k)} - \theta_a^{(k-1)}\right\| > \epsilon$ **do**
  Update $\mu^{(k)} = B\theta^{(k)}$ via $\mu^{(k)} = H(y - B_a\theta_a^{(k-1)})$, $H = B(B^TB + \lambda R)^{-1}B^T$
  Update $\theta_a^{(k)}$ by $\theta_a^{(k)} = S_{\frac{\gamma}{2}}(B_a^T(y - \mu^{(k)}))$
  Update $k = k + 1$
end

The fact that both subproblems have closed-form solutions significantly speeds up the optimization algorithm. A summary of the BCD algorithm for the orthogonal case is given in Algorithm 1. Although the BCD algorithm, in general, may not converge to an optimum, even if the function is convex [91], the following proposition guarantees that the BCD attains the global optimum of problem (A.1).

Proposition 2. *The BCD algorithm attains the global optimum of the SSD loss function in* (A.1).

The proof is given in Appendix B.

General Basis B_a

For the general basis B_a, we first show that the SSD problem can be reduced to a weighted LASSO problem via Proposition 3.

Proposition 3. *The SSD problem in* (A.1) *is equivalent to a weighted LASSO problem in the form of*

$$\arg\min_{\theta_a} F(\theta_a) = (y - B_a\theta_a)^T(I - H)(y - B_a\theta_a) + \gamma\|\theta_a\|_1 \tag{3.2}$$

with $H = B(B^TB + \lambda R)^{-1}B^T$.

The proof is given in Appendix C.

Common LASSO solvers such as least angle regression (LARS) [92] and quadratic programming [89] cannot solve the above weighted LASSO problem for high-dimensional data. For example, for an image of the size 350 by 350, i.e., $p \approx 10^5$, the LARS algorithm [92] will find the entire solution path in about 60 hours, which is impractical for real-time purposes. Alternatively, we develop an efficient algorithm based on the accelerated proximal gradient method [93] for solving the large scale optimization problem in (A.2).

The proximal gradient (PG) method is a class of optimization algorithms focusing on minimization of the summation of a group of convex functions, some of which are non-differentiable. The function $F(\theta_a)$ in (A.2), is comprised of $f(\theta_a) = (y - B_a\theta_a)^T(I - H)(y - B_a\theta_a)$, which is convex differentiable if R is a positive semi-definite matrix (see Appendix D for the proof of convexity) and $g(\theta_a) = \gamma\|\theta_a\|_1$, which is a non-differentiable function. Another assumption of the PG algorithm is that the continuous part of the objective function $f(\theta_a)$ (A.1) is convex differentiable with the Lipschitz continuous gradient L, i.e., there exists a constant L that for every $\alpha, \beta \in \mathbb{R}$,$\|\nabla f(\alpha) - \nabla f(\beta)\| \leq L\|\alpha - \beta\|$, where $\nabla f(\cdot)$ is the gradient function. As shown in Appendix E, $f(\theta_a)$ is Lipschitz continuous gradient with $L = 2\|B_a\|_2^2$. Provided the above assumptions, the PG method optimizes $F(\theta_a)$ through an iterative algorithm given by $\theta_a^{(k)} = \text{argmin}_{\theta_a}\{f(\theta_a^{(k-1)}) + \left\langle \theta_a - \theta_a^{(k-1)}, \nabla f(\theta_a^{(k-1)}) \right\rangle + \frac{L}{2}\|\theta_a - \theta_a^{(k-1)}\|^2 + \gamma\|\theta_a\|_1\}$, where super-indices (k) and $(k-1)$ denote iteration numbers and $\langle \cdot, \cdot \rangle$ is the inner product operator. For more details on proximal gradient method, readers can refer to [94]. Through the following proposition, we show that in each iteration PG results in a closed-form solution for the SSD problem in the form of a soft-thresholding function.

Proposition 4. *The proximal gradient method for the SSD problem in* (A.1)*, given by* $\theta_a^{(k)} = \text{argmin}_{\theta_a}\{f(\theta_a^{(k-1)}) + \left\langle \theta_a - \theta_a^{(k-1)}, \nabla f(\theta_a^{(k-1)}) \right\rangle + \frac{L}{2}\|\theta_a - \theta_a^{(k-1)}\|^2 + \gamma\|\theta_a\|_1\}$*, has a closed-*

Algorithm 2: Optimization algorithm for solving SSD based on APG

initialize

$L = 2\|B_a\|_2^2$, $\theta_a^{(0)} = 0$, $x^{(0)} = 0$, $t_0 = 1$, $k = 1$

end

while $\left\|\theta_a^{(k)} - \theta_a^{(k-1)}\right\| > \epsilon$ **do**

Let $\mu^{(k)} = H(y - B_a\theta_a^{(k-1)})$

Update $\theta_a^{(k)} = S_{\frac{\gamma}{L}}(x^{(k-1)} + \frac{2}{L}B_a^T(y - B_a x^{(k-1)} - \mu^{(k)}))$

Update $t_k = \frac{1+\sqrt{1+4t_{k-1}^2}}{2}$

Update $x^{(k)} = \theta_a^{(k-1)} + \frac{t_{k-1}-1}{t_k}(\theta_a^{(k-1)} - \theta_a^{(k-2)})$

Update $k = k + 1$

end

form solution in each iteration k, in the form of a soft-thresholding function as follows:

$$\theta_a^{(k)} = S_{\frac{\gamma}{L}}(\theta_a^{(k-1)} + \frac{2}{L}B_a^T(y - B_a\theta_a^{(k-1)} - \mu^{(k)})) \tag{3.3}$$

with $L = 2\|B_a\|_2^2$.

The proof is given in Appendix F.

The soft-thresholding solution provided by PG can significantly expedite the SSD implementation and anomaly detection. Suppose B_a is of size $n \times k_a$, and B is of size $n \times k_\mu$. The most computationally expensive operator in the soft-thresholding solution is $\theta_a^{(k-1)} + \frac{2}{L}B_a^T(y - B_a\theta_a^{(k-1)} - \mu^{(k)})$. So, the total computational complexity in each iteration is around $k_\mu^3 + 6n^2k_\mu$ flops, which is quadratic in n.

To increase the convergence speed of the proximal gradient method, [95] showed that with the adjustment of the step size, it is possible to achieve the quadratic convergence rate $O(\frac{1}{k^2})$. With this adjustment, the proposed optimization algorithm for solving SSD based on the accelerated proximal gradient (APG) algorithm is summarized in Algorithm 2.

It is worth noting that if B_a is orthogonal ($B_a^T B_a = I$), the PG algorithm is reduced to the BCD algorithm.

3.1.2 Generalization to 2-D image case

In the previous section, we discussed SSD for one-dimensional cases and presented efficient algorithms for performing SSD. In this section, we extend our formulation and algorithms to two-dimensional images. Suppose a 2-D image $Y_{n_1 \times n_2}$ is available. We define B_i and $B_{a,i}$; $i = 1, 2$ as the basis for the background and anomalous regions, respectively, where i ($i = 1, 2$) denotes the basis in the x and y direction of the image. Therefore, the tensor product of these 1-D bases (i.e., $B = B_1 \otimes B_2$) can give the proper 2-D basis for the background as well as anomalous regions. Consequently, the SSD problem for the 2-D case can be written as

$$\underset{\theta, \theta_S}{argmin} \, \|\tilde{e}\|^2 + \lambda \theta^T R \theta + \gamma |\theta_a|_1, \; s.t. \; \tilde{y} = B\theta + B_a \theta_a + \tilde{e}, \tag{3.4}$$

where $B = B_2 \otimes B_1$, $B_a = B_{a,2} \otimes B_{a,1}$, $\tilde{y} = \mathrm{vec}(Y)$, $\tilde{e} = \mathrm{vec}(e)$, $\otimes$ is the tensor product, and $\mathrm{vec}(\cdot)$ is an operator that unfolds a matrix to a column vector. Note that the size of the resulting basis is defined by the size of their individual bases - for example, if the size of matrix B_i is $n_i \times k_{\mu_i}$, in which k_{μ_i} is the number of basis in the i^{th} direction, then the size of B is $n_1 n_2 \times k_{\mu_1} k_{\mu_2}$. Similarly, assume that $B_{a,i}$ is of size $n_i \times k_{a_i}$, then B is an $n_1 n_2 \times k_{a_1} k_{a_2}$matrix.

To solve the SSD problem in (3.4), we can still use algorithms presented in 1-D cases. However, since both APG and BCD algorithms require matrix inversion operations to compute the projection matrix $H = B^T (B^T B + \lambda R)^{-1} B$, and the computational complexity of the matrix inversion is nonlinearly proportional to the size of $(B^T B + \lambda R)$, i.e., $O\left((k_{\mu_1} k_{\mu_2})^3\right)$, the complexity of the APG or BCD algorithm is given by a sixth order polynomial of $(n_1, n_2, k_{\mu_1}, k_{\mu_2}, k_{a_1}, k_{a_2})$ with the leading term $(k_{\mu_1}^3 k_{\mu_2}^3 + 6 n_1^2 n_2^2 k_{\mu_1} k_{\mu_2})$. Consequently, this becomes computationally intractable as the size of the image increases. To reduce the computational complexity, we define matrix R in such a way that matrix H can be computed by a tensor product of two low-dimensional matrices. Following [96], we define

matrix R as $R = B_2^T B_2 \otimes D_1^T D_1 + D_2^T D_2 \otimes B_1^T B_1 + \lambda D_2^T D_2 \otimes D_1^T D_1$, which results in a decomposable projection matrix, i.e., $H = H_2 \otimes H_1$, where $H_i = B_i(B_i^T B_i + \lambda D_i^T D_i)^{-1} B_i^T$, with the dimensions of $k_i; i = 1, 2$. D_i is the first order difference matrix in i^{th} dimension. This trick makes the algorithm very efficient for 2-D images, as it requires the inversion of matrices with lower dimensions, i.e., $B_i^T B_i + \lambda D_i^T D_i$. Hence, the computational complexity of the matrix inversion operation is reduced from $O\left((k_{\mu_1} k_{\mu_2})^3\right)$ to $O\left(k_{\mu_1}^3 + k_{\mu_2}^3\right)$.

Since $(P \otimes Q)\theta = \mathrm{vec}(P\Theta Q)$ with Θ as the matrix form of θ, updating steps in Algorithms 1 and 2 can be shown in the matrix form as $\mu^{(k)} = H_1(B_{a,1}\Theta_a^{(k-1)} B_{a,2}^T - Y)H_2$, $\Theta_a^{(k)} = S_{\frac{\gamma}{2}}(B_a^T(Y - \mu^{(k)}))$ for BCD and $\Theta_a^{(k)} = \mathrm{S}_{\frac{\gamma}{2}}(X^{(k-1)} + \frac{2}{L} B_{a,1}^T(Y - B_{a.1} X^{(k-1)} B_{a,2}^T - \mu^{(k)}) B_{a,2})$ for APG. The total computational complexity is computed by a third-order polynomial of $(n_1, n_2, k_{\mu_1}, k_{\mu_2}, k_{a_1}, k_{a_2})$, with the leading term $(k_{\mu_1}^3 + k_{\mu_2}^3 + 6n_1^2 k_{\mu_1} + 6n_2^2 k_{\mu_2} + 2(n_1 + n_2)k_{a_1} k_{a_2} + 2n_1 n_2(n_1 + n_2) + 2n_1 n_2(k_{a_1} + k_{a_2}))$, which is computationally more efficient.

3.1.3 Choice of tuning parameters λ and γ

In the SSD model, two tuning parameters, λ and γ, are used to control the smoothness of the background $\hat{\mu}$ and the sparsity of anomalous regions $\hat{a}$, respectively. A common approach for choosing these tuning parameters is to use the *k*-fold cross-validation method on a 2-D grid of parameters (λ, γ) and find the pair of parameters that minimizes the mean squared error. However, this approach requires solving the SSD problem for each pair of (λ, γ) on the gird, which may not be feasible. Alternatively, we propose an iterative approach that updates the tuning parameters in each iteration of the APG and BCD algorithms without exploring all pairs of (λ, γ).

We begin with some initial values for the tuning parameters. In the k^{th} iteration, along with $\theta^{(k)}$, λ is also updated based on the general cross-validation (GCV) criterion as follows: $\lambda^{(k)} = \arg\min_\lambda \mathrm{GCV}(\lambda) = \arg\min_\lambda \frac{\|Y - H_1(\lambda)(Y - a^{(k-1)})H_2(\lambda) - a^{(k-1)}\|^2/n}{(1 - n^{-1}\mathrm{tr}(H^{(k)}(\lambda))^2}$ [29], where $a^{(k)} = B_{a,1} X^{(k-1)} B_{a,2}^T$. As $H_i(\lambda)$ that involves matrix inversion should be computed for

different values of λ in each iteration, we use a series of transformations and operations inspired by [97] to increase the computational speed. We, first, calculate the Cholesky decomposition of $B_i^T B_i$ that gives the square matrix Z_i. Then, using the eigenvalue decomposition, we calculate the eigenvalues and eigenvectors of matrix $Z_i^{-1} D_i^T D_i (Z_i^{-1})^T$. That is, $U_i \mathrm{diag}(\mathrm{s_i}) U_i^T = Z_i^{-1} D_i^T D_i (Z_i^{-1})^T$. Next, the calculated eigenvectors are used to define matrix $V_i = B_i (Z_i^{-1})^T U_i$, which is calculated prior to optimization. Therefore, in each iteration, $H_i(\lambda)$ can be computed by $H_i(\lambda) = V_i^T \mathrm{diag}(\frac{1}{1+\lambda s_1}, \cdots, \frac{1}{1+\lambda s_n}) V_i$. As can be seen, the calculation of $H_i(\lambda)$ does not involve matrix inversion, which makes its computation much more efficient. The detailed derivation is shown in Appendix G.

To select the tuning parameter $\gamma^{(k)}$ in the k^{th} iteration, the GCV criterion can still be used. However, GCV usually tends to select more pixels, leading to a larger false positive rate. This is because GCV is a function of the residual sum of square (RSS) in the following way: $GCV = \frac{RSS/n}{(1-n^{-1}Tr(H))^2}$. However, in the case of anomaly detection, the goal is to precisely identify the anomalous regions rather than to achieve a smaller RSS. Therefore, we utilize the Otsus method [43] for finding $\gamma^{(k)}$. Otsu shows that minimizing the intra-class variance $\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$ is the same as maximizing inter-class variance $\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t)\left[\mu_1(t) - \mu_2(t)\right]^2$, where weights $\omega_i(t)$ are the probabilities of the two classes separated by a threshold t and σ_i^2 are the variances of these classes. In our case, classes are defined as the background and the anomalous regions. Both $\omega_i(t)$ and $\mu_i(t)$ can easily be computed from the histogram $p(j)$ by $\omega_1(t) = \sum_{j=0}^{t} p(j)$, $\omega_2(t) = 1 - \omega_1(t)$, $\mu_1(t) = [\sum_{j=0}^{t} p(j)x(j)]/\omega_1$ and $\mu_2(t) = [\sum_{j=t+1}^{n} p(j)x(j)]/\omega_2$. For simplicity, we use otsu(y) to represent the function that returns $\gamma^{(k)}$ by applying Otsus method on y. For detailed information on Otsus method, see [43].

The detailed optimization algorithm with parameter selection is shown in Algorithm 3.

Algorithm 3: Optimization algorithm for solving SSD based on APG and BCD with tuning parameter selection

initialize

Choose the basis for background as B_1, B_2 and anomalies as $B_{a,1}, B_{a,2}$, respectively

$Z_i Z_i^T = B_i^T B_i$, $U_i \mathrm{diag(s_i)} U_i^T = Z_i^{-1} D_i^T D_i (Z_i^{-1})^T$, $V_i = B_i (Z_i^{-1})^T U_i$, $i = 1, 2$

$H_i(\lambda) = V_i^T (I + \lambda \mathrm{diag}(s_i))^{-1} V_i$, $i = 1, 2$ is a function of λ

$L = 2\|B_{a,1}\|_2^2 \|B_{a,2}\|_2^2$,

$\Theta_a^{(0)} = 0$, $X^{(0)} = 0$, $t_1 = 1$, $k = 1$

end

while $\left\| \Theta_a^{(k-1)} - \Theta_a^{(k)} \right\| > \epsilon$ **do**

Select $\lambda^{(k)} = \mathrm{argmin}_\lambda \, GCV(\lambda) = \arg\min_\lambda \frac{\|Y - H_1(\lambda)(Y - A^{(k-1)})H_2(\lambda) - A^{(k-1)}\|^2 / n}{(1 - n^{-1}\mathrm{tr}(\hat{H}(\lambda))^2}$

$H_i^{(k)} = H_i(\lambda^{(k)})$, $i = 1, 2$

$A^{(k)} = B_{a,1} X^{(k-1)} B_{a,2}^T$, $M^{(k)} = H_1^{(k)} (Y - A^{(k-1)}) H_2^{(k)}$

$\Theta_e^{(k)} := X^{(k-1)} + \frac{2}{L} B_{a,1}^T (Y - M^{(k)} - A^{(k)}) B_{a,2}$

(In BCD algorithm, for orthogonal $B_{a,i}, i = 1, 2$, $\Theta_e^{(k)} := B_{a,1}^T (Y - M^{(k)}) B_{a,2}$)

Select $\gamma^{(k)}$ by Otsu's method $\gamma^{(k)} = \mathrm{otsu}(\Theta_e^{(k)}) \times L$

$\Theta_a^{(k)} = S_{\frac{\gamma}{L}}(\Theta_e^{(k)})$

Update $t_k = \frac{1 + \sqrt{1 + 4t_{k-1}^2}}{2}$

Update $X^{(k)} = \Theta_a^{(k-1)} + \frac{t_{k-1} - 1}{t_k} (\Theta_a^{(k-1)} - \Theta_a^{(k-2)})$

Update $k = k + 1$

end

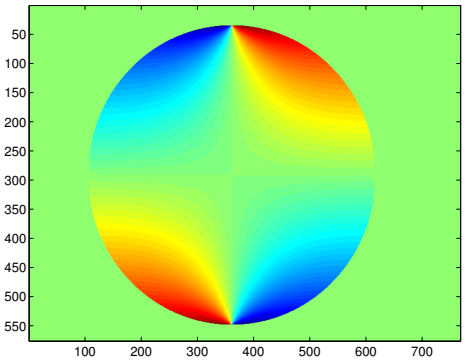
3.1.4 Choice of basis for background and anomalous regions

Another important factor in the implementation of SSD is the type of basis chosen for the background and anomalous regions. In this section, we provide some general guidelines for basis selection. As it is assumed that the background is smooth, any smooth basis, such as splines or kernels, can be used for the background. If a spline basis is used, the number of knots is another parameter that should be chosen. As pointed out by [97], as long as the number of knots is sufficiently large to capture the variation of the background, a further increase in knots will have little effect due to the regulation of smoothness via λ. [97] proposed using the GCV criterion to select the right number of knots. A similar approach can be used to select the number of knots for the background, assuming that anomalies are only a small part of the entire image.

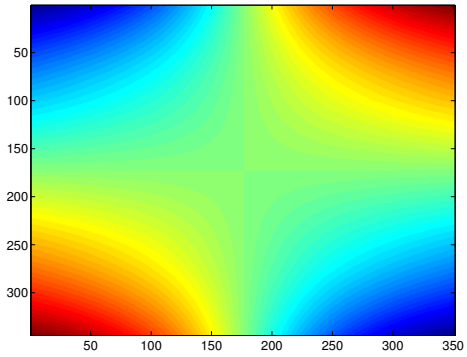
Selecting the basis to better represent anomalous regions is a more challenging task, and prior information about the size and shape of anomalies would be useful for choosing a suitable basis. For example, if anomalies are small regions scattered over the background or are in the form of thin lines, then it is recommended to use an identity basis, i.e., $B_a = I$. However, if the anomalies form clustered regions, a spline basis can be a better choice. For example, if anomalous regions have sharp corners (e.g., a rectangular shape), a 0-order or linear B-splines will suffice. For regions with curved boundaries, quadratic or cubic B-spline bases are recommended. Prior information about the size of anomalies, if available, could help in the selection of the right number of knots for a spline basis. In general, a smaller number of knots may result in the loss of detection accuracy, and a larger number of knots will lead to the selection of normal regions with large noises.

3.2 Simulation study

In this section, the performance of the proposed Smooth-Sparse Decomposition method is evaluated through simulations under different conditions. Specifically, we consider three



(a) Theoretical principal stress direction of a loaded circle

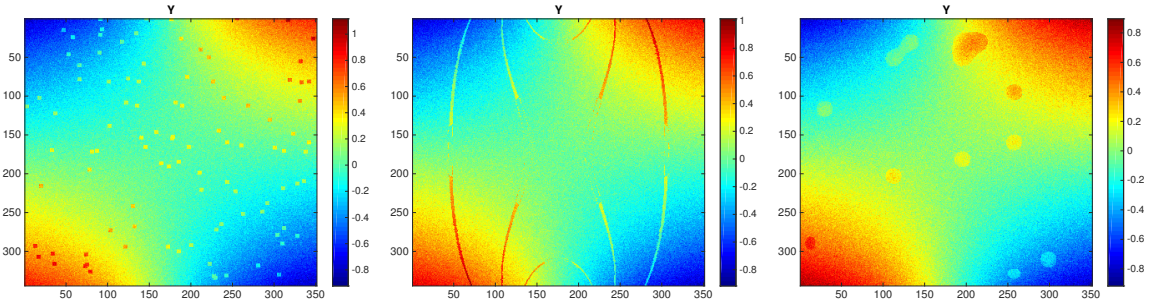


(b) Background

Figure 3.1: Simulated background from principal stress direction of a loaded circle

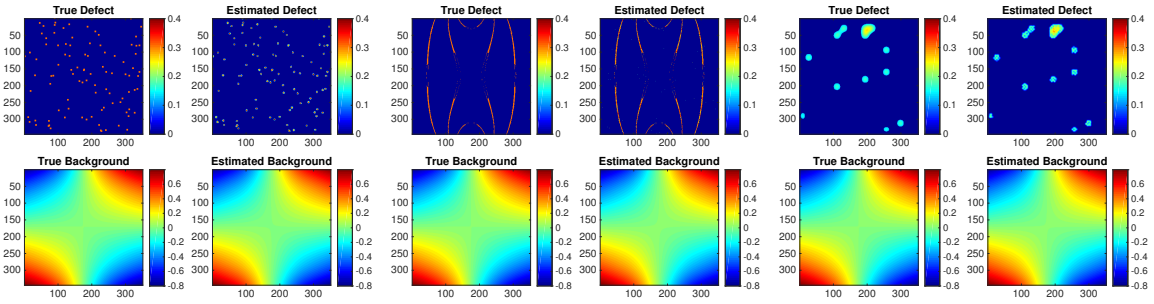
different types of anomalies: the scattered type, in which anomalies are some random pixels in the image, the line type, in which anomalies are represented by several thin lines, and the clustered type, in which the clusters of anomalies are spread over the image. A sample of these generated image with different types of anomalies is shown in Figure 3.2. We simulate a 350×350 image Y according to the following model $Y = M + A + E$, in which M is the true background, A represents anomalous regions, and E is the random noise. The smooth background M (see Figure 3.1b) is obtained from the photoelasticity experiments from the center part of the theoretical stress direction of a loaded circle where the load is applied to its top and bottom points (see Figure 3.1a). The anomalies are generated by $A = \delta \cdot I(a \in A_s)$, in which A_s is the set of anomalous pixels, and δ characterizes the intensity difference between anomalies and the background, which is set to be 0.3. For the scattered case, A_s is defined based on randomly generated 5×5 squares. For the line case, a set of line-shaped systematic anomalies caused by numerical errors in photo-elasticity experiments is used. For the clustered case, anomalies are randomly generated clusters comprising about 341 pixels. We generate the random noise E by $E_i \sim NID(0, \sigma^2)$ with $\sigma = 0.05$.

We compare our proposed SSD method with four existing methods in the literature that follow the two-step approaches described in the introduction. The benchmark algorithms



(a) The simulated image of scattered anomaly (b) The simulated image of line anomaly (c) The simulated image of clustered anomaly

Figure 3.2: The Simulated image of scattered and clustered anomalies



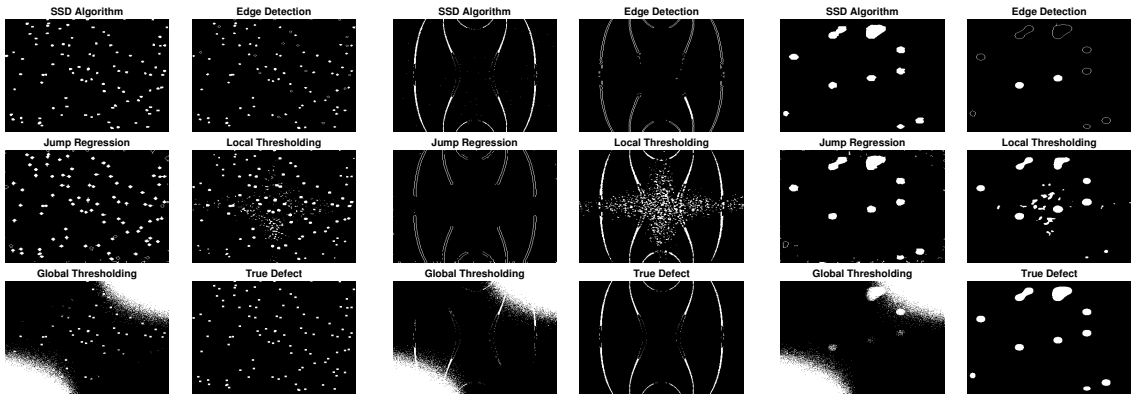
(a) Result of the proposed method in scattered anomaly (b) Result of the proposed method in line anomaly (c) Result of the proposed method in clustered anomaly

Figure 3.3: SSD decomposition results for scattered, line and clustered anomalies

which we used for comparison include Sobel edge detection [36], jump regression with local polynomial kernel regression [98], the Otsu global thresholding method [43], and the Nick local thresholding method [99]. As per reviewer's suggestion, we also compare our method with the extended-maxima transformation method. The comparison results are reported in the Supplementary Material. For Sobel edge detection, Otsu global thresholding method and Nick local thresholding method, a smoothing [100] is first applied as preprocessing step to remove the noises. For the edge-based methods namely, the Sobel edge detection and jump regression model, edge thinning algorithm, edge linking algorithms and filling algorithms [48] are applied to close the boundary and fill the area inside the boundary.

For the proposed SSD method, we use the identity matrix, cubic B-spline basis with 175×175 knots and cubic B-spline basis with 85×85 knots in the case of line anomalies, scattered anomalies, and clustered anomalies for anomaly basis B_a, respectively. The cubic B-spline basis B with 7×7 knots is chosen for the background. The tuning parameters λ and γ are selected automatically based on the GCV criterion and Otsu's method as discussed in Section 3.1.3.

To evaluate the performance of the proposed methodology and benchmark methods, we repeat the simulation procedure 100 times and the following criteria are calculated and compared: false positive rate (FPR), defined as the proportion of normal pixels predicted as anomaly; and false negative rate (FNR), defined as the proportion of anomalous pixels predicted as normal; background recovery square root mean square error (e_μ), defined as the square root of mean square error of the background estimator $\hat{\mu}$: $e_\mu = \sqrt{\|\mu - \hat{\mu}\|^2}$; anomalies recovery square root mean square error (e_a), defined as the square root of mean square error of the anomalies estimator $\hat{a}$: $e_a = \sqrt{\|a - \hat{a}\|^2}$; and the computation time. The FPR, FNR, and computational time of all methods for all scenarios are reported in Table 3.1. Since other benchmark method cannot give the estimation of the background and anomalies, only e_μ and e_a of the proposed SSD is reported in Table 3.2. Detected



(a) Result of benchmark method in scattered anomaly

(b) Result of benchmark method in line anomaly

(c) Result of benchmark method in clusterd anomaly

Figure 3.4: Detected anomalies for scattered, line, and clustered cases

regions along with true anomalies for one of the simulation replications with $\delta = 0.3$ is shown in the binary plots in Figure 3.4.

From Figure 3.4 and Table 3.1, it can be seen that in terms of the FPR and FNR, our SSD method overall exhibits a better performance than other benchmark methods. For example, in the line case, The FPR and FNR of SSD are 0.001 and 0.003, respectively, which are the least among all methods. The edge detection method has a lower FPR than SSD in the clustered case and scattered case. However, it shows an FNR of 0.754 in the case of clustered Anomalies which is much larger than that of SSD, 0.001. The reason for such a high FNR is that the edge-based method only detects the boundaries and often fails to give closed-boundary regions even with edge linking and filling algorithm applied. Therefore, the inner sections of anomalous regions are not detected as seen in Figure 3.4b and 3.4c. Jump regression, on the other hand, can detect the edge more precisely by the local kernel polynomial regression, thus reduce the FNR drametically. Global thresholding has the worst performance with FPR and FNR values around 0.20 and 0.50 in all cases. The poor performance of this method is because that it can only identify the anomalies globally with one single thresholding value. In contrast, local thresholding has better FPR and FNR values than global thresholding techinques, since it identifies the anomalous regions

Table 3.1: FPR, FNR, and computation time for line anomalies, clustered anomalies and scattered anomalies with $\delta = 0.3$

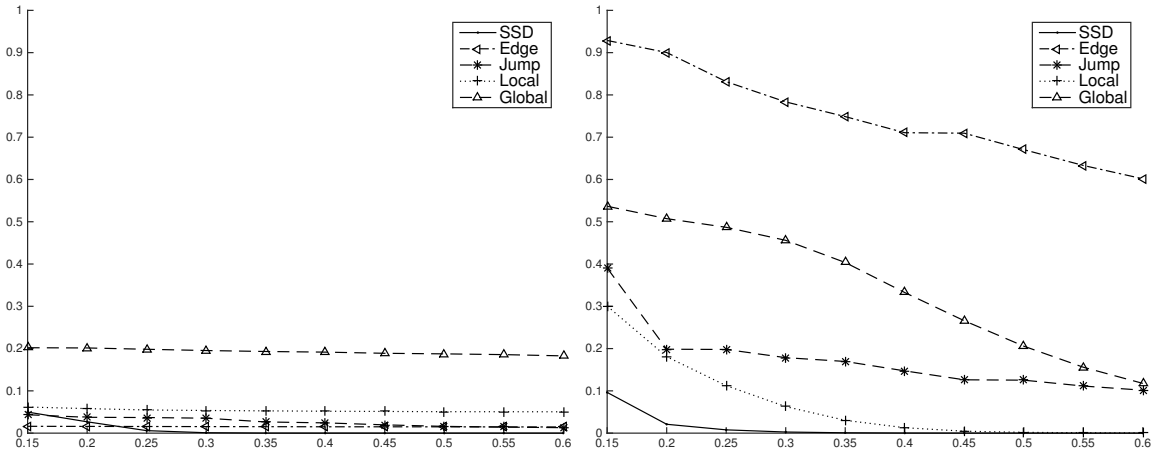
	Line Anomalies			Clustered Anomalies		
	FPR	FNR	Time	FPR	FNR	Time
SS Decomposition	0.001	0.003	0.129s	0.018	0.001	0.19s
Edge Detection	0.015	0.783	0.945s	0.001	0.754	0.409s
Jump Regression	0.035	0.111	38.43s	0.081	0.054	37.736s
Local Thresholding	0.054	0.063	0.043s	0.046	0.289	0.045s
Global Thresholding	0.195	0.456	0.046s	0.211	0.572	0.048s
	Scattered Anomalies					
	FPR	FNR	Time			
SS Decomposition	0.012	0.007	0.267s			
Edge Detection	0.003	0.257	0.667s			
Jump Regression	0.11	0.063	37.796s			
Local Thresholding	0.02	0.087	0.045s			
Global Thresholding	0.203	0.407	0.048s			

Table 3.2: The square root of mean square error of background and anamalies estimator with $\delta = 0.3$

	Line Anomalies		Clustered Anomalies		Scattered Anomalies	
	e_μ	e_a	e_μ	e_a	e_μ	e_a
SS Decomposition	$3.7e-4$	$6e-4$	$3.6e-4$	$1.5e-3$	$4.2e-4$	$7e-4$

directly by thresholding the image locally. However, this method is sensitive to noises, and hence falsely detecting more anomalies in normal regions than the SSD method. For example, in the clustered case, the FPR of the local thresholding method is 0.289 which is almost 300 times larger than that of SSD, 0.001. In terms of computation time, local and global thresholding methods as well as SSD have comparable computational times. The edge detection is slow because it entails post-processing steps. Jump regression has the highest computation time as it requires fitting local polynomial kernel regression models for each pixel.

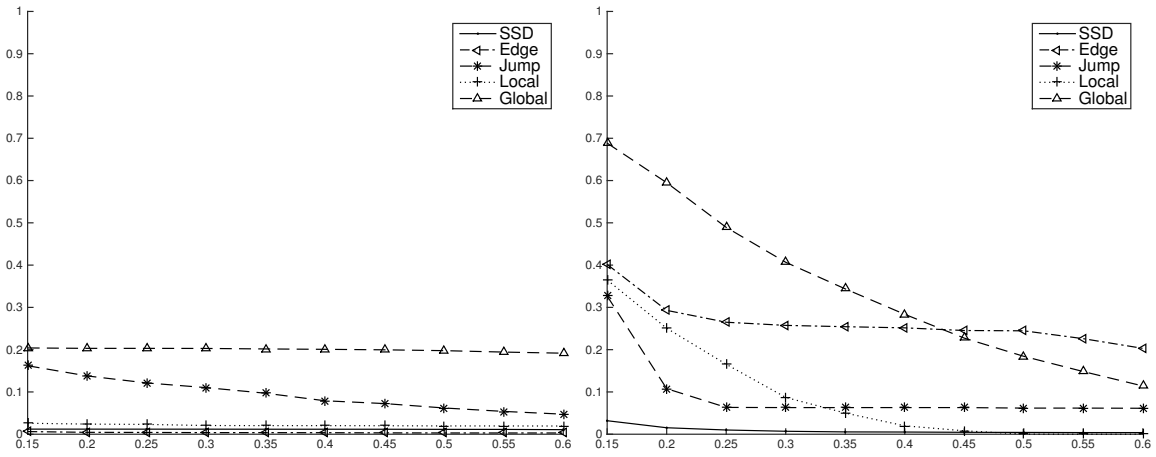
In order to study the sensitivity of these algorithms, we also run a similar simulation setting for different anomaly magnitudes δ. FPR and FNR values of all methods for all cases are reported in Figures 3.5 - 3.7. In terms of FNR, one can see that as δ increases, the



(a) Result of FPR for different δ (b) Result of FNR for different δ

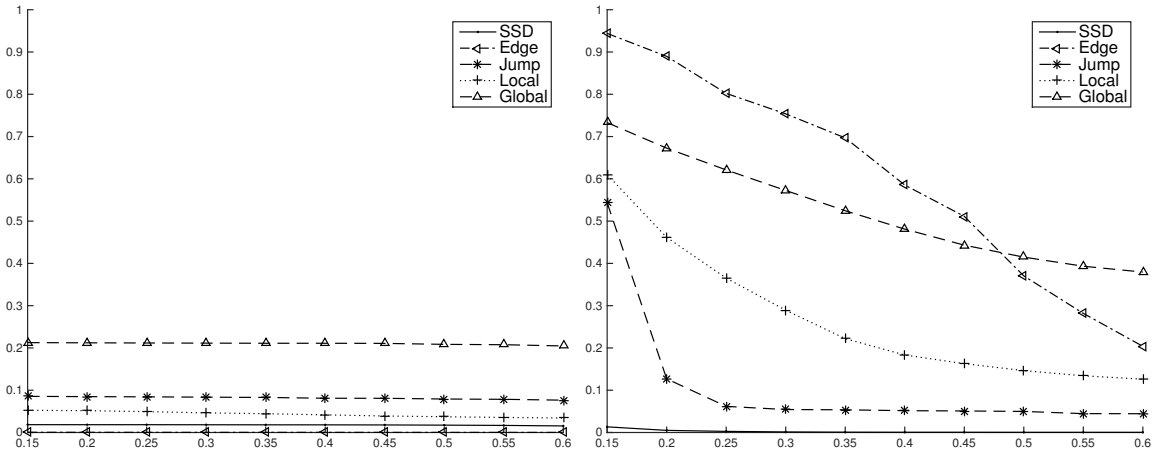
Figure 3.5: Sensitivity study for line anomaly

FNR of the SSD methods converges to 0 much faster than the rest. In addition, local thresholding and jump regression have much smaller FNR value than other benchmark methods. This indicates that if the intensity contrast between anomalous regions and the background is large, both of these methods will accurately detect all the regions. Jump regression has smaller performance (smaller FNR) especially in the clustered case indicating that it identifies the jump location precisely between the clustered defect and the background. However, in the case of line defect, due to the difficulty to close the boundary, it has larger FNR rate than local thresholding. Both global thresholding and Sobel edge detection have very large FNR rate even δ is large. In terms of FPR, both local thresholding and jump regression have larger FPR than SSD methods implying that it selects some noisy pixels as anomaly. Global thresholding has much larger FPR. Sobel Edge Detection sometimes has a very small FPR at the cost of a very large FNR. Furthermore, the comparison of the FNR and FPR plots for different methods indicates that our comparative observations in the previous simulation with $\delta = 0.3$ is valid for other δ values. In short, considering the reported error rates and computation time, the proposed SSD method overall outperforms other benchmark methods.



(a) Result of FPR for different δ (b) Result of FNR for different δ

Figure 3.6: Sensitivity study for scattered anomaly



(a) Result of FPR for different δ (b) Result of FNR for different δ

Figure 3.7: Sensitivity study for clustered anomaly

3.3 Case study

In this section, the proposed SSD method is applied to a case study of anomaly detection and feature extraction in the area of photo-elasticity, a non-destructive evaluation method. Photo-elasticity is a non-contact optical technique developed based on the birefringence property exhibited by translucent materials. This method can acquire maximum shear stress (isochromatics) and principal stress direction (isoclinics) by recording the transmitted light through translucent materials. Unlike point-by-point techniques such as X-ray diffraction, ultrasonic test, and micro-Raman spectroscopy, photo-elasticity is able to obtain the full-field stress quantification directly from the digital camera, making it popular in the stress analysis of various manufacturing products, such as silicon wafers and translucent composite laminates.

The setup of photo-elasticity experiments used in this chapter is shown in Figure 5.7. In this setup, a near infra-red light source of wavelength 1150nm and a digital camera equipped with a low-pass filter was used to record necessary images. Two polarizers are placed on both sides of the specimen, one between the source and the specimen and the other between the specimen and the analyzer. Four images are taken based on the different angles of the two polarizers. Two quarter plates are then added to both sides of the specimen between the specimen and the polarizers. After that, six images are taken by changing the angles of two polarizers and quarter-wave plates. Finally, using Maxwell's stress optic law, the maximum shear stress is obtained from these ten images to quantify the stress distribution in the specimen [101].

A multi-crystalline bending silicon beam and a silicon surface laminate with surface indentation [103] were inspected using the photo-elasticity experiments as described above. The stress maps of these samples are shown in Figures 1.2a and 1.2b. In the silicon beam, we are interested in extracting the "grain boundaries." Grain boundaries used in crystallography represent the interface between two crystallines, which is often in the form of

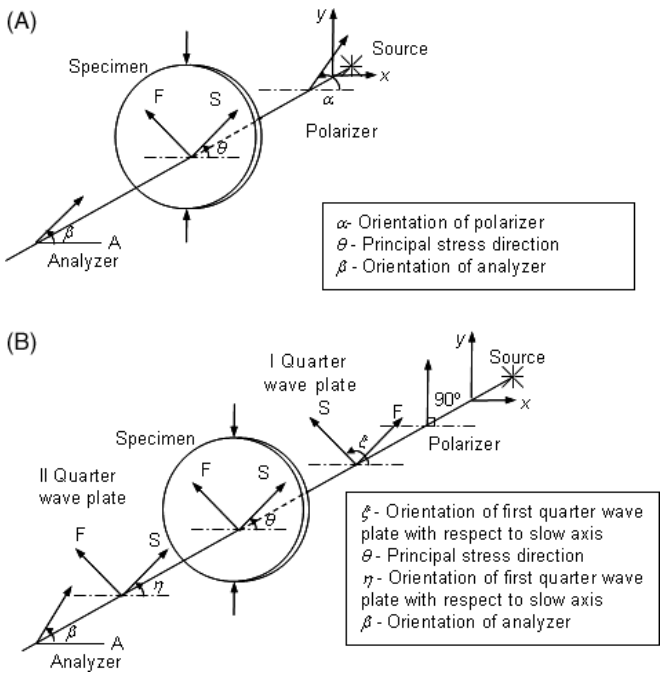


Figure 3.8: Photo-elasticity experiment setup [102]

Table 3.3: Case Study Sample Description

Sample	Description	Image Size	Defect	Defect Type
Sample 1	Multi-crystalline Silicon	200 by 2910	Grain Boundaries	Line
Sample 2	Silicon Surface	90 by 550	Surface Indentation	Clustered

swerving lines. In the silicon surface sample, high-stress areas indicate the surface indentation important to be detected in quality inspection. These indentations often form clusters of high-stress areas. A summary of the sample specifications is given in Table 3.3.

We applied the proposed SSD method and other benchmarks on these stress maps to separate the anomalous regions (i.e., grain boundaries and indentations) from the background of stress maps. In SSD, we used an identity basis for detecting grain boundaries and a cubic B-spline basis with 24×139 knots for detecting the indentations. We also used a cubic B-spline basis with knots 21×21 and 11×51 for the backgrounds of these two samples, respectively. Detected regions from each sample are shown in Figures 3.9a and 3.9b. As can be seen in Figure 3.9a, the SSD algorithm gives a clearer representation of the grain boundaries than other benchmarks. The edge-detection method is sensitive to

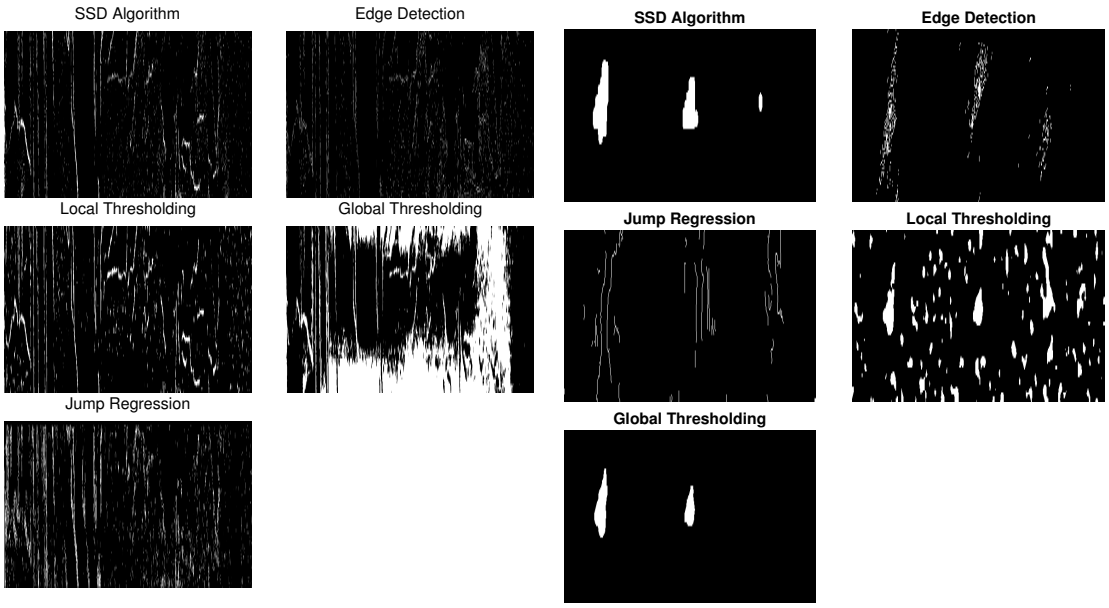
Table 3.4: Computational time for all methods

Sample	SSD	Edge Detection	Local Thresholding	Global Thresholding	Jump Regression
Sample 1	0.350s	6.44s	0.320s	0.270s	$534.05s$
Sample 2	0.034s	0.543s	0.030s	0.020s	50.069s

noise and hence, misses a significant portion of the grain boundaries. Global thresholding performs poorly for Sample 1, since the range of the background intensity in this sample is large. Detected boundaries by local thresholding and jump regression resemble those by SSD. From Figure 3.9b, it can be seen that the SSD algorithm can identify all three indentation clusters with no false detection. As the background has a smaller intensity range, the detection performance of global thresholding is better than its performance in Sample 1. However, it still fails to detect the small indentation region because it thresholds the entire image with a single thresholding value without considering the local smoothness. Local-thresholding and edge-detection methods are sensitive to noise, which leads to larger detection errors. Although jump regression can detect all three indentations, its FPR is high. The computation times for each of these methods are also reported in Table 3.4. Threshold-based methods and SSD have comparable computation times, whereas edge detection and jump regression are significantly slower. For example, the computation time of SSD in Sample 2 is around 0.034 seconds, 16 times faster than the edge-detection method. The difference in the computation times between Samples 1 and 2 is because of the image size.

3.4 Conclusion

Image data are increasingly used for online inspection and anomaly detection in various manufacturing and service applications. In this chapter, we proposed Smooth-Sparse Decomposition for image denoising and anomaly detection. Unlike existing methods, which perform denoising and detection separately, SSD is a one-step approach that is able to model and separate the background, anomalies and defect. This method improve both the



(a) Detected anomalies in Sample 1 (b) Detected anomalies in Sample 2

Figure 3.9: Detection results of Sample 1 and Sample 2 for all methods

detection accuracy and computation time under the smooth background with various defect types. We formulated the SSD problem in the form of high-dimensional regression augmented with penalty terms to encourage both smoothness of background and sparsity of anomalies in a certain basis. To efficiently solve the large-scale optimization problem for SSD, we used BCD and APG methods and proposed efficient iterative algorithms that have closed-form solutions in each iteration. We also proposed an iterative method for the quick selection of tuning parameters. Using simulations, the performance of proposed SSD was evaluated and compared with some existing methods in terms of detection accuracy and computation time for three types of anomalies. Based on simulation results, we concluded that, overall, the proposed SSD algorithm outperforms other benchmarks. We further showed that the error rate of the proposed SSD algorithm converges to 0 as the anomaly intensity increases. Other benchmarks did not show this property. Additionally, to demonstrate how the proposed method can be applied to real data, we analyzed the stress maps obtained from the photo-elasticity experiments by using SSD. In the case study, we

used the stress images of a silicon beam and surface sample that possessed different types of anomalies. We showed that the SSD algorithm can identify anomalous regions precisely in both samples.

The main focus of this chapter was to provide a framework for one-step, real-time, and automatic algorithms for anomaly detection under smooth background. One extension is to generalize SSD for other types of backgrounds like textured backgrounds. To achieve this, one may use other types of basis such as Fourier basis, wavelet basis, and kernel can in the SSD model.

CHAPTER 4

REAL-TIME MONITORING AND DIAGNOSIS OF HIGH-DIMENSIONAL FUNCTIONAL DATA STREAMS VIA SPATIO-TEMPORAL SMOOTH SPARSE DECOMPOSITION

In this chapter, we present a process monitoring technique that effectively deals with high dimensional streaming data and is able to consider different spatio-temporal structure of functional mean and anomalies. The method extends the smooth sparse decomposition (SSD) into spatio-temporal SSD, which decomposes the original tensor stacked by sequential profiles or images into three parts: namely, the smooth spatio-temporal correlated functional mean, anomalies, and random noises. Furthermore, we propose two temporal models, reproducing kernel models and roughness minimization models to model the temporal trend of the system. A recursive updating procedure of those two models is also proposed for real-time monitoring applications.

The remainder of Chapter 4 is organized as follows. Section 4.1 elaborates the proposed spatio-temporal SSD for HD data stream. In Section 4.2, reproducing kernel and roughness penalization are used for temporal modeling and recursive estimation procedures are proposed for real-time analysis. In Section 4.3, monitoring and diagnosis methods are proposed by combining LRT with spatio-temporal SSD. To evaluate and compare the proposed methodology with existing methods, simulated data based on thermodynamic principles of heat transfer are used in Section 4.4. In Section 4.5, we illustrate how our proposed method can be used in real world using three case studies including monitoring of a rolling process, detection of solar flares, and monitoring of a forging process. We conclude the chapter in Section 4.6.

4.1 Spatio-Temporal Smooth Sparse Decomposition

In this section, we develop the spatio-temporal model by extending SSD so it can model the temporal trend in addition to the spatial structure of functional data streams. We also propose efficient algorithms for fast implementation of spatio-temporal SSD (ST-SSD) for a given data sample. For simplicity, we begin with profile data (i.e. 1D functional data). Suppose a sequence of profiles y_t; $t = 1, \cdots, n$ is available where y_t is a profile of size $p \times 1$ recorded at time t. We combine all profiles into a matrix $Y = (y_1, y_2, \cdots, y_n)$ of size $p \times n$ and define $y = vec(Y)$ as the vectorized matrix (i.e., y is a $pn \times 1$ vector). Following [9], we aim to decompose y into three components: A functional mean μ, anomalies a, and noises e as $y = \mu + a + e$, where $a = vec(a_1, \cdots, a_n)$ and $e = vec(e_1, \cdots, e_n)$ with a_t and e_t as anomaly features and noise in y_t. We assume that the dynamic functional mean μ has a smooth spatio-temporal structure and a is sparse or can be sparsely represented by a certain basis. To model both spatial and temporal structures and at the same time reduce data dimensions, we define B_s and B_t as smooth spatial and temporal bases for the mean, and B_{as} and B_{at} as spatial and temporal bases for anomalies, respectively. The spatio-temporal bases for the mean and anomalies are obtained by the tensor product of these bases, i.e., $B = B_t \otimes B_s$ and $B_a = B_{at} \otimes B_{as}$. Consequently, the functional mean and anomalies are modeled as $\mu = (B_t \otimes B_s)\theta$ and $a = (B_{at} \otimes B_{as})\theta_a$ resulting in $y = (B_t \otimes B_s)\theta + (B_{at} \otimes B_{as})\theta_a + e$, where $\theta = vec(\theta_1, \theta_2, \cdots, \theta_n)$ and $\theta_a = vec(\theta_{a,1}, \cdots, \theta_{a,n})$, and θ_t and $\theta_{a,t}$ are the spatio-temporal coefficients of the functional mean and anomalies at time t, correspondingly. We assume that noise components are normally independently distributed i.e., $e \sim NID(0, \sigma^2)$. To estimate θ and θ_a, we propose a penalized regression model, called spatio-temporal smooth sparse decomposition (ST-SSD), as follows:

$$\underset{\theta, \theta_a}{argmin} \, \|e\|^2 + \theta^T R \theta + \gamma \|\theta_a\|_1 \text{ s.t. } y = (B_t \otimes B_s)\theta + (B_{at} \otimes B_{as})\theta_a + e. \quad (4.1)$$

where $\|\cdot\|$ and $\|\cdot\|_1$ are L_2 and L_1 norm operators, and γ is a tuning parameter to be determined by the user. The Matrix R is the regularization matrix that controls the smoothness of the mean function, and the L_1 penalty term, $\gamma|\theta_a|_1$, encourages the sparsity of the anomalous regions. In this chapter, inspired by [96], we define the regularization matrix R as $R = R_t \otimes B_s^T B_s + B_t^T B_t \otimes R_s + R_t \otimes R_s$, where R_s and R_t are the regularization matrices that control the smoothness in the spatial and temporal directions. For tensors with smooth structure, it has shown in [96] and [9] that the penalty term defined with this tensor structure is able to achieve high precision with small computational time and asymptotically achieve the optimal rate of convergence under some mild conditions. The spatial regularization matrix R_s can be defined as $R_s = \lambda_s D_s^T D_s$ [104], where D_s is the first order difference matrix since the smoothness of the function is directly related to the difference between the neighbor coefficients. That is, $D_s = [d_{pq}] = 1_{q=p} - 1_{q=p+1}$, with 1_A as an indicator function i.e., it is 1 when A is true, and 0 otherwise. λ_s is the tuning parameter controlling the spatial smoothness of the functional mean. The choice of R_t depends on the temporal model and will be discussed in Section 4. It is shown in [9] that if θ_a is given, $\mu = B\theta$ can be solved by $\mu = H(y - B_a\theta_a)$, where $H = B(B^T B + R)^{-1} B^T$ is the projection matrix. They also showed that (4.1) is equivalent to a weighted lasso formulation, i.e., $\min_{\theta_a}(y - B_a\theta_a)^T(I - H)(y - B_a\theta_a) + \gamma\|\theta_a\|_1$, thus can be efficiently solved by the APG algorithm. The reason for defining the regularization matrix in the foregoing form is that under this definition of R, the projection matrix of ST-SSD, denoted by H, can be further decomposed by the tensor product of two spatial and temporal projection matrices, i.e., $H = H_t \otimes H_s$, where $H_s = B_s(B_s^T B_s + R_s)^{-1} B_s^T$ and $H_t = B_t(B_t^T B_t + R_t)^{-1} B_t^T$, as shown in Appendix A. This will help significantly reduce the computational complexity of the optimization algorithm for solving Equation (4.1). Equation (4.1) is a convex optimization problem that can be solved via a general convex solver such as the interior point method. However, the interior point method is slow and cannot be used in HD settings. Therefore, similar to [9], the accelerated proximal gradient (APG) algorithm is used

Algorithm 4: Optimization algorithm for solving SSD

initialize

$L = 2\|B_{as}\|_2^2$, $x^{(0)} = 0$, $\theta_a^{(0)} = 0$, $t_0 = 1$

end

Compute

$$H_s = B_s(B_s^T B_s + R_s)^{-1} B_s^T$$

$$H_t = B_t(B_t^T B_t + R_t)^{-1} B_t^T \tag{4.2}$$

for $k = 1, 2, \cdots$ **do**

Update

$$a^{(k-1)} = (B_{at} \otimes B_{as}) x^{(k-1)}$$

$$\mu^{(k-1)} = (H_t \otimes H_s)(y - a^{(k-1)}) \tag{4.3}$$

$$\theta_a^{(k)} = S_{\frac{\gamma}{L}}(x^{(k-1)} + \frac{2}{L}(B_{at}^T \otimes B_{as}^T)(y - a^{(k-1)} - \mu^{(k-1)})) \tag{4.4}$$

$$t_k = \frac{1 + \sqrt{1 + 4t_{k-1}^2}}{2}$$

$$x^{(k)} = \theta_a^{(k)} + \frac{t_{k-1} - 1}{t_k}(\theta_a^{(k)} - \theta_a^{(k-1)})$$

if $\left|\theta_a^{(k)} - \theta_a^{(k-1)}\right| < \epsilon$ **then**

Stop

end

end

to solve (4.1) iteratively, as given in Algorithm 4.

In Algorithm 4, $S_\gamma(x) = \mathrm{sgn}(x)(|x| - \gamma)_+$ is a soft-thresholding operator, in which $\mathrm{sgn}(x)$ is the sign function and $x_+ = \max(x, 0)$. The θ is not explicitly update since it is updated with μ as $\mu^{(k)} = B\theta^{(k)}$. Note that the convergence of Algorithm 4 is guaranteed and can be proved similarly as shown in [9].

To generalize the ST-SSD model to l-dimensional data (e.g., $l = 2$ for images or multi-channel signals), we represent a single sample by tensor $\mathcal{Y}$ of size $p_1 \times \cdots \times p_l$. For computational efficiency, the spatial basis B_s and B_{as} are defined as the tensor product of multiple

Algorithm 5: Optimization algorithm for solving SSD based on APG

initialize

$\Theta_a^{(0)} = 0$, $\mathcal{X}^{(0)} = 0$, $t_0 = 1$

$L = 2\prod_i \|B_{ai}\|_2^2$

$H_{si} = B_{si}(B_{si}^T B_{si} + R_{si})^{-1} B_{si}^T$, $i = 1, \cdots, k$

$H_t = B_t(B_t^T B_t + R_t)^{-1} B_t^T$

end

for $k = 1, 2, \cdots$ **do**

Update $\mathcal{A}^{(k-1)} = \mathcal{X}^{(k-1)} \times_{i=1}^{k} B_{si} \times_t B_{st}$

$\mathcal{M}^{(k)} = (\mathcal{Y} - \mathcal{A}^{(k-1)}) \times_{i=1}^{k} H_{si} \times_t H_t$

$\Theta_a^{(k)} = S_{\frac{\gamma}{L}}(\mathcal{X}^{(k-1)} + \frac{2}{L}(\mathcal{Y} - \mathcal{A}^{(k-1)} - \mathcal{M}^{(k-1)}) \times_{i=1}^{k} B_{si}^T \times_t B_{st}^T)$

$t_k = \frac{1+\sqrt{1+4t_{k-1}^2}}{2}$

$\mathcal{X}^{(k)} = \Theta_a^{(k)} + \frac{t_{k-1}-1}{t_k}(\Theta_a^{(k)} - \Theta_a^{(k-1)})$

if $|\Theta_a^{(k-1)} - \Theta_a^{(k)}| < \epsilon$ **then**

Stop

end

end

1D bases, i.e., $B_s = \otimes_{i=1}^{l} B_{si}$ and $B_{as} = \otimes_{i=1}^{l} B_{asi}$, where $\otimes_{i=1}^{l} B_{si} := B_{s1} \otimes \cdots \otimes B_{sl}$. It is shown in appendix A that if we set $R_s = \otimes_{i=1}^{l}(B_{si}^T B_{si} + R_{si}) - B_s^T B_s$, the projection matrix becomes decomposable, that is $H_s = \otimes_{i=1}^{l} H_{si}$ with $H_{si} = B_{si}(B_{si}^T B_{si} + R_{si})^{-1} B_{si}^T$. For example, if a B-spline basis is used, R_{si} can be defined as $R_{si} = \lambda_{si} D_i^T D_i$. Furthermore, to increase the computational efficiency of the optimization algorithm, we use the well-known relationship between the Kronecker and tensor products to compute $y = (\otimes_{i=1}^{l} B_{si})x$ by $\mathcal{Y} = \mathcal{X} \times_{i=1}^{l} B_{si} := \mathcal{X} \times_1 B_{s1} \times_2 B_{s2} \cdots \times_l B_{sl}$, in which $\mathcal{X} \times_n B_{sn}$ is the n-mode tensor product defined by $(\mathcal{X} \times_n B_{sn})(i_1, \cdots, i_l) = \sum_{j_n} \mathcal{X}(i_1, \cdots, j_n, \cdots, i_l) B_{sn}(i_n, j_n)$. A summary of the optimization algorithm for solving the generalized ST-SSD problem is given in Algorithm 5. In this algorithm, since the matrix inversion can be performed in each dimension separately, i.e., $B_{si}^T B_{si} + R_{si}; i = 1, \cdots, l$, the total complexity of the matrix inversion is reduced from $O\left(n^3 \prod_i k_i^3\right)$ to $O\left(n^3 + \sum_i k_i^3\right)$, assuming B_{si} is of size $p_i \times k_i$.

Selection of an appropriate basis for the functional mean and anomaly is important to model the spatio-temporal structure of a data stream. Therefore, due to its computa-

tional efficiency and flexibility, the B-Spline basis is commonly used for modeling nonlinear smooth functions. In this chapter, for the spatial basis, we assume that the functional mean is smooth and can be modeled with B-spline basis. Selecting a basis for anomalous regions depends on the type of anomalies. For example, if anomalies are randomly scattered over the mean, it is recommended to use an identity basis, i.e., $B_{as} = I$. If anomalies form clustered regions, a spline basis can be a better choice. More details about the spatial basis selection of the functional mean and anomalies are given in [9]. We also assume that anomalies appear abruptly, and hence they do not have a specific temporal structure. Therefore, we use the identity matrix as the temporal basis for anomalies, i.e., $B_{at} = I$. In the following section, we will discuss the choice of temporal basis for the functional mean and the recursive estimation of ST-SSD.

4.2 ST-SSD for Streaming Data and Recursive Estimation

The proposed ST-SSD can effectively model both the temporal and spatial structure of HD data streams. However, the estimation method given in Algorithm 4 is only efficient for a given data stream with a fixed number of observations, n. In the context of statistical process control (SPC), process monitoring includes two stages known as Phase I and Phase II. Since the functional mean is unknown in the beginning, we use n in-control (IC) observations in Phase I to learn the distribution of the monitoring statistic and the control limit. The baseline control chart estimated in Phase I, can then be used for real-time and online monitoring in Phase II. Therefore, the proposed method can be used to conduct Phase I analysis offline on n observations collected offline. However, for online (phase II) analysis of HD data where streaming samples are being recorded in short sampling intervals, Algorithm 4 with the complexity of $O\left(n^3 + \sum_i k_i^3\right)$ loses its efficiency over time as n grows linearly by time. Specifically, when a new sample is recorded at time t, the length of y increases by the dimensions of the recorded data. Consequently, after some time, the dimensions of Problem (4.1) become so large that it cannot be solved by any optimization

algorithms. To address this issue, the key idea is to develop a recursive estimation procedure that only requires the previous estimations and current data to solve the optimization problem. This recursive algorithm significantly reduces the computation time and required memory, which enables real-time implementation of the method. For this purpose, we use special temporal bases for the functional mean, B_t, and penalization term, R_t. In the following subsections, we propose two temporal models based on reproducing kernels and roughness minimization and present a recursive estimator for each model.

4.2.1 Reproducing Kernels

Reproducing Kernel Hilbert Space (RKHS) is a functional space widely used for modeling smooth functional forms using kernels [105]. From the representer theorem [106], it is known that any function in an RKHS can be written as a linear combination of kernel functions evaluated at time t. Hence, the gram matrix K_t, defined as $(K_t)_{ij} = \kappa(i,j)$ $(i,j = 1, \cdots, t)$, can be used as the temporal basis (i.e. $B_t = K_t$) in (4.1), where $\kappa(i,j)$ is the kernel function. In this chapter, we use the Gaussian kernel to model the smooth temporal structure defined as $\kappa(i,j) = \exp(-\frac{(i-j)^2}{2c^2})$ [107], where c is the bandwidth of the Gaussian kernel). To control the smoothness of the temporal trend, we use Hilbert norm penalization [106], which is equivalent to defining $R_t = \lambda_t K_t$ in Equation (4.1). λ_t is the tuning parameter controlling the temporal smoothness of the functional mean. Consequently, the projection matrix H_t can be computed by

$$H_t = K_t(K_t^2 + \lambda_t K_t)^{-1} K_t = K_t K_{\lambda_t,t} \tag{4.5}$$

where $K_{\lambda_t,t} = (K_t + \lambda_t I_t)^{-1}$. However, since computing (4.5) requires inversion of $K_t + \lambda_t I_t$, which is an $t \times t$ matrix, the total complexity is $O(t^3)$ at time t. Eventually, computing (4.5) is not feasible due to the increasing number of observations and the limited computational resources. To reduce the computational complexity, we propose to solve the estimation recursively with only recent w observations since earlier observations typically

have little impact on the current estimation. We define $K_t = \kappa(i,j) \quad i,j = t-w+1, \cdots, t$ and $\tilde{K}_t = \kappa(i,j) \quad i,j = t-w, \cdots, t$ as windowed kernel functions, and define $K_{\lambda_t,t} = (K_t + \lambda_t I)^{-1}$ and $\tilde{K}_{\lambda_t,t} = (\tilde{K}_t + \lambda_t I)^{-1}$, accordingly. Proposition 1 shows that H_t and $K_{\lambda_t,t}$ can be computed recursively.

Proposition 5. *The following update rules hold for H_t and $K_{\lambda_t,t}$*

$$\tilde{H}_t = \begin{bmatrix} \tilde{H}_{t-1} - k_{t-1} r_{t-1}^T g_{t-1}(I_{t-1} - \tilde{H}_{t-1}) & (I_{t-1} - \tilde{H}_{t-1}) k_{t-1} g_{t-1} \\ r_{t-1}^T (I_{t-1} + k_{t-1} r_{t-1} g_{t-1} - g_{t-1}) & (1 - r_{t-1}^T k_{t-1}) g_{t-1} \end{bmatrix} \tag{4.6}$$

$$\tilde{K}_{\lambda_t,t} = \begin{bmatrix} K_{\lambda,t-1} + r_{t-1} r_{t-1}^T g_{t-1} & -r_{t-1} g_{t-1} \\ -r_{t-1}^T g_{t-1} & g_{t-1} \end{bmatrix}$$

where $r_t = \tilde{K}_{\lambda_t,t} k_t$, $H_t = \tilde{H}_t(2:t, 2:t)$, $K_{\lambda_t,t} = \tilde{K}_{\lambda_t,t}(2:t, 2:t)$ $k_t = [\kappa(t-w+1,t), \cdots, \kappa(t-1,t)]^T$, $g_{t-1} = (1 + \lambda_t - r_{t-1}^T k_{t-1})^{-1}$.

$\tilde{K}_{\lambda_t,t}(2:t, 2:t)$ denotes the reduced matrix $\tilde{K}_{\lambda_t,t}$ after removing the first row and column of the matrix. The proof of Proposition 1 is given in Appendix B. With this recursive updating rule, it is not hard to show that the total complexity of Algorithm 4 will reduce to $O(w^2)$ at each sampling time t, which is more efficient compared to the non-recursive case with $O(t^3)$. The fact that the complexity does not grow with the rate of $O(t^3)$ enables the real-time implementation of ST-SSD for online monitoring of HD streaming data. Finally, the optimization (estimation) algorithm can be updated by replacing the computation of the projection matrix H_t in Algorithm 4 with the updating procedure in (4.6).

4.2.2 Roughness Minimization

In this section, we propose an alternative approach for temporal modeling that can achieve even faster computational speed than reproducing kernels. In cases where the functional mean is less volatile over time, we suggest a simple temporal basis and roughness matrix, namely, $B_t = I_t$ and $R_t = D_t^T D_t$ in (4.1), in which D_t is the first order difference matrix

of size $(t-1)\times t$ defined as $D_t = [d_{pq}] = 1_{q=p} - 1_{q=p+1}$. By choosing R_t to be $D_t^T D_t$, the temporal penalization term, $\theta^T R_t \theta = \theta^T D_t^T D_t \theta = \sum_{i=2}^{t} \|\theta_i - \theta_{i-1}\|^2$, becomes roughness penalization that penalizes the first order difference of θ_t for a smoother estimation over time. Therefore, the temporal projection matrix is given by

$$H_t = (I_t + \lambda_t D_t^T D_t)^{-1}. \tag{4.7}$$

The next step is to design a recursive estimator for the roughness minimization model. As mentioned earlier, for a system with a gradual temporal trend, it is often true that recent observations have more impact and therefore are more important for parameter estimation and updating. Therefore, an approximate, yet accurate, approach is to estimate only the most recent coefficient θ_t without changing the previous estimations of $\theta_1, \cdots, \theta_{t-1}$. This is equivalent to solve (4.1) for only θ_t and $\theta_{a,t}$. In this way, the ST-SSD model in (4.1) can be reduced to the following model, which only requires the estimation of θ_t and $\theta_{a,t}$ at time t.

$$\underset{\theta_t, \theta_{a,t}}{argmin} \|e\|^2 + \theta^T R \theta + \gamma \|\theta_a\|_1, \ \text{ subject to } y_t = B_s \theta_t + B_{as} \theta_{a,t} + e_t, \tag{4.8}$$

where $R = I_t \otimes R_s + \lambda_t D_t^T D_t \otimes B_s^T B_s + \lambda_t D_t^T D_t \otimes R_s$. As shown in Proposition 2, given $\theta_{a,t}$ and previous estimates, θ_t has a closed-form solution.

Proposition 6. *Suppose the previous estimation $\hat{\theta}_1, \cdots, \hat{\theta}_{t-1}, \hat{\theta}_{a,1}, \cdots, \hat{\theta}_{a,t-1}$ and $\hat{\theta}_{a,t}$ are known, then the solution of θ_t (or equivalently $\mu_t = B_s \theta_t$) to* (4.8) *is given by*

$$\hat{\mu}_t = B_s \hat{\theta}_t = (1 - \tilde{\lambda}_t) \hat{\mu}_{t-1} + \tilde{\lambda}_t H_s (y_t - \hat{a}_t), \tag{4.9}$$

where $\tilde{\lambda}_t = \frac{1}{1+\lambda_t}$ and $\hat{a}_t = B_{at} \hat{\theta}_{a,t}$

The proof is shown in Appendix C. Note that in (4.9), the temporal structure of μ_t is modeled by the weighted average of the previous estimation $\hat{\mu}_{t-1}$ and the current estimation

of $H_s(y_t - \hat{a}_t)$, which is a recursive equation similar to the monitoring statistic of the EWMA control chart. Therefore, for a stationary process, $\hat{\mu}_t$ can help average the noise over time, which leads to a stationary distribution with a much smaller variance than the original data. However, different from the EWMA control chart, we use (4.9) to estimate the true dynamic trend $\hat{\mu}_t$ in dynamic processes. The spatial structure of μ_t is captured by applying the projection matrix H_s. However, $\hat{\theta}_{a,t}$ (or equivalently $\hat{a}_t = B_{at}\hat{\theta}_{a,t}$) is unknown and should also be estimated. To efficiently solve for $\theta_{a,t}$, we first show that the loss function is equivalent to a weighted lasso formulation, which can be solved via an accelerated proximal gradient algorithm.

Proposition 7. *Suppose the previous estimation* $\hat{\theta}_1, \cdots, \hat{\theta}_{t-1}, \hat{\theta}_{a,1}, \cdots, \hat{\theta}_{a,t-1}$ *are known, then Problem* (4.8) *is equivalent to the following weighted lasso formulation:*

$$\min_{\theta_{a,t}} F(\theta_{a,t}) = \min_{\theta_{a,t}} (y_t - B_{as}\theta_{a,t})^T (I - \tilde{\lambda}_t H_s)(y_t - B_{as}\theta_{a,t}) - 2(1-\tilde{\lambda}_t)(y_t - B_{as}\theta_{a,t})^T y_{t-1} + \gamma \|\theta_{a,t}\|_1 \tag{4.10}$$

where $\tilde{\lambda}_t = \frac{1}{1+\lambda_t}$.

The proof is given in Appendix D. To efficiently solve this weighted lasso formulation, we propose to use the proximal gradient method, which is a class of optimization algorithms focusing on minimization of the summation of a group of convex functions, some of which are non-differentiable. The function $F(\theta_{a,t})$ in (A.2), is comprised of a differentiable convex function $f(\theta_{a.t}) = (y_t - B_{as}\theta_{a,t})^T (I - \tilde{\lambda}_t H_s)(y_t - B_{as}\theta_{a,t}) - 2(1 - \tilde{\lambda}_t)(y_t - B_{as}\theta_{a,t})^T y_{t-1}$ and a non-differentiable L_1 penalty $g(\theta_a) = \gamma \|\theta_{a,t}\|_1$. It can be proved that the proximal gradient algorithm converges to a global optimum given R_s is a positive semi-definite matrix. This is true because $f(\theta_{a,t})$ is convex and Lipchitz continuous (see Appendix E for the proof of convexity and Appendix F for the proof of Lipchitz continuity.) According to the following proposition, the proximal gradient method leads to a closed-form solution for $\theta_{a,t}$ in each iteration of the optimization algorithm.

Proposition 8. *The proximal gradient problem for* (A.2)*, given by* $\theta_{a,t}^{(k)} = \text{argmin}_{\theta_{a,t}} \{ f(\theta_{a,t}^{(k-1)}) +$

Algorithm 6: Recursive algorithm for roughness minimization

initialize

$\theta_a^{(0)} = 0$, $L = 2\|B_{as}\|_2^2$, $t_0 = 1$, $x_t^{(0)} = 0$

$H_s = B_s(B_s^T B_s + R_s)^{-1} B_s^T$

end

for each time t

while $|\theta_{a,t}^{(k-1)} - \theta_{a,t}^{(k)}| > \epsilon$ **do**

Update

$a_t^{(k-1)} = B_{as} x_t^{(k-1)}$

$\mu_t^{(k-1)} = (1 - \tilde{\lambda}_t)\hat{\mu}_{t-1} + \tilde{\lambda}_t H_s(y_t - a_t^{(k-1)})$

$\theta_{a,t}^{(k)} = S_{\frac{\gamma}{L}}(\theta_{a,t}^{(k-1)} + \frac{2}{L} B_{as}^T(y_t - a_t^{(k-1)} - \mu_t^{(k-1)}))$

$t_k = \frac{1+\sqrt{1+4t_{k-1}^2}}{2}$

$x_t^{(k)} = \theta_{a,t}^{(k)} + \frac{t_{k-1}-1}{t_k}(\theta_{a,t}^{(k)} - \theta_{a,t}^{(k-1)})$

end

$\left\langle \theta_{a,t} - \theta_{a,t}^{(k-1)}, \nabla f(\theta_a^{(k-1)}) \right\rangle + \frac{L}{2}\|\theta_{a,t} - \theta_{a,t}^{(k-1)}\|^2 + \gamma\|\theta_{a,t}\|_1\}$, *has a closed-form solution in each iteration* k*, in the form of a soft-thresholding function as follows:*

$$\theta_{a,t}^{(k)} = S_{\frac{\gamma}{L}}(\theta_{a,t}^{(k-1)} + \frac{2}{L} B_{as}^T(y_t - B_{as}\theta_{a,t}^{(k-1)} - \mu_t^{(k)})), \tag{4.11}$$

where $L = 2\|B_{as}\|_2^2$.

The proof is given in appendix G. Finally, by combining the estimator from both (4.9) and (A.3), Problem (4.8) can be solved iteratively and recursively with the accelerated proximal gradient algorithm as shown in Algorithm 6. The accelerated proximal gradient algorithm is an *accelerated* version of the proximal gradient (PG) algorithm, which is able to achieve a better convergence rate than the PG algorithm.

4.2.3 ST-SSD for Stationary Processes

In a stationary process where the functional mean of the data stream is constant when the process is in-control, ST-SSD is simplified by removing the temporal basis of the mean, i.e., $\mu = B_s\theta$. Hence, Equation (4.8) becomes $\operatorname{argmin}_{\theta_t,\theta_{a,t}} \|e\|^2 + \theta^T R\theta + \gamma\|\theta_a\|_1, \ \text{subject to} y_t = B_s\theta + B_{as}\theta_{a,t} + e_t$, which can be solved by Algorithm 6 with a slight modification in es-

timating μ. As the functional mean is constant, the temporal projection matrix reduces to a sample average function. Consequently, the functional mean in Algorithm 6 is estimated by $\hat{\mu}^{(k)} = H_s(\frac{1}{n}\sum_{i=1}^{n}(y_i - a_i^{(k-1)}))$. It is noteworthy that the ST-SSD model for stationary processes is a special case of the roughness minimization model with $\lambda_t \to \infty$ and the kernel model with $c \to 0$. More detailed discussions are given in Appendix H.

4.3 Online Process Monitoring and Diagnostics

In this section, we propose a monitoring procedure that combines the ST-SSD model with a sequential likelihood ratio test. We also discuss how ST-SSD can be used for diagnosis after a change is detected.

4.3.1 Construct Monitoring Statistics

We propose an online monitoring method using the estimated sparse anomalous features from ST-SSD. If the sparse vector of anomalies detected by ST-SSD, i.e., $\hat{a}$ is statistically significant, it can be implied that a process change has occurred. In this chapter, we focus on two types of temporal changes: the first type, studied in the simulation study, is based on the change-point model where the anomaly appears after a time point τ. In the second type, discussed in the case study, the anomaly happens only in short-time windows. It should be noted that in both cases, the anomaly is non-smooth in the temporal domain due to the sudden jump. We denote the detected anomaly at time t as $\hat{a}_t$. Therefore, at each time *t*, we test whether the expected residuals after removing the functional mean, denoted by $\mu_{r,t}$, is zero or has a mean shift in the direction of $\hat{a}_t$. That is,

$$H_0 : \mu_{r,t} = 0 \quad vs \quad H_1 : \mu_{r,t} = \delta\hat{a}_t;\ \delta > 0.$$

To test these hypotheses, a likelihood ratio test is applied to the residuals at each sampling time t, i.e., $r_t = y_t - \mu_t$. This leads to the test statistic $T_\gamma(t) = \frac{(\hat{a}_t^T r_t)^2}{\hat{a}_t^T \hat{a}_t}$ [108], in which it is

assumed that the residuals r_t are independent after removing the functional mean and their distribution before and after the change remains the same.However, the test statistics $T_\gamma(t)$ relies on the selection of γ since it directly controls the sparsity of $\hat{a}_t$. To construct a more stable hypothesis test, inspired by [109], we develop a monitoring statistic by combining multiple tuning parameters. [109] proposed to use different values of the tuning parameter γ obtained from the breakpoints of the piecewise linear solution path of LASSO. This is a very time consuming process. For example, for an images stream with the size of 350×350, the LARS algorithm finds the entire solution path in about 60 hours, which makes it impractical for real-time monitoring purposes. Consequently, we use a smaller set of possible tuning parameters denoted by Γ_{n_γ}. It is known that when γ is large enough, i.e., $\gamma \geq \gamma_{max}$, every element of coefficient θ_a will become 0. Therefore, we define the set of tuning parameter γ as $\Gamma_{n_\gamma} = \{\frac{\gamma_{\max} i}{n_\gamma} | i = 0, 1, \cdots, n_\gamma\}$ by dividing $(0, \gamma_{max}]$ equally into n_γ intervals. The choice of γ_{max} is discussed in the next subsection. Thus, the combined test statistic can be defined as

$$\tilde{T}(t) = \max_{\gamma \in \Gamma_{n_\gamma}} \frac{T_\gamma(t) - E(T_\gamma(t))}{\sqrt{\mathrm{Var}(\tilde{T}_\gamma(t))}} \tag{4.12}$$

where $E(T_\gamma(t))$ and $\mathrm{Var}(\tilde{T}_\gamma(t))$ respectively are the mean and variance of $T_\gamma(t)$ under H_0, that are estimated using a set of in-control data. An out-of-control sample is detected when its corresponding monitoring statistic $\tilde{T}(t)$ is greater than a control limit h.

4.3.2 Control Limit Determination

The value of the control limit is computed based on a predetermined in-control average run length (or equivalently, type I error rate) and the set of the tuning parameter values, Γ_{n_γ}. [109] suggested to determine Γ_{n_γ} by using the least angle regression (LARS) algorithm [92] that provides the entire solution path. The breakpoints in such a solution path define the set Γ_q. However, the complexity of the LARS algorithm with p covariates is $O(np+p^3)$,

which is infeasible for HD data. Alternatively, to define Γ_q, we use equidistant values of γ within a certain range. The procedure for computing the control limit h in Phase I analysis using an in-control sample of HD is summarized as follows: First, a ST-SSD algorithm such as Algorithm 4 (for 1D profile) or Algorithm 5 (for image or high-dimensional tensor) is applied to an in-control sample $Y = (y_1, y_2, \cdots, y_n)$ to estimate μ and a. The parameters λ_s and λ_t are tuned via the GCV criterion as proposed in [3] and the kernel bandwidth c are selected by using the cross validation criterion. Next, the set of tuning parameter is defined by $\Gamma_{n_\gamma} = \{\frac{\gamma_{max} i}{n_\gamma} | i = 0, 1, \cdots, n_\gamma\}$, γ_{max} is determined such that $\theta_a = 0$ for all the IC samples. Larger values of n_γ increase the detectability of the monitoring procedure. However, if too large, the monitoring procedure becomes computationally inefficient. In this chapter, based on numerical experiments, we found that for $n_\gamma \geq 20$ the detection power in detecting small shifts are similar. Therefore, we use $n_\gamma = 20$ in this chapter. After that, Similar to [109], assuming the dynamic mean can be estimated accurately (this is validated in the simulation study), we generate *i.i.d* gaussian random draws to simulate the residuals r_t. We then apply the ST-SSD on the simulated data, compute the monitoring statistics $\tilde{T}(t)$, and estimate its empirical distribution. Finally, the control limit is determined as a certain quantile of the empirical distribution of the monitoring statistics based on a predetermined IC average run length.

4.3.3 Diagnosis of Detected Changes

After the proposed control chart triggers an out-of-control signal, the next step is to diagnose the detected change. Diagnosis for functional data is defined by determining portions of data that have a different structure from the functional mean. In many cases, especially in the HD setting, estimating the location of anomalies responsible for the out-of-control signal is important. This information would help process engineers identify and eliminate the potential root causes. Suppose that the control chart triggers a signal at time τ, we then apply the LRT test procedure described in the previous section to determine which γ pro-

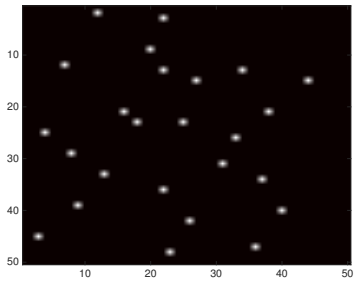
vides the largest test statistics in (4.12), denoted by $j^* = \arg\max_{j=1,\cdots,q_t} \frac{T_{\gamma_j}(\tau) - E(T_{\gamma_j}(\tau))}{\sqrt{\mathrm{Var}(\tilde{T}_{\gamma_j}(\tau))}}$. Vector $\hat{a}_{\gamma_{j^*},\tau} = B_{as}\hat{\theta}_{a\tau}$ is the estimated anomalies for the optimal γ_{j^*} at time τ. Since a localized basis (e.g. band matrix) is used for B_{as}, the sparsity of $\hat{\theta}_{a\tau}$ leads to the sparsity of $\hat{a}_{\gamma_{j^*},\tau}$. Therefore, the non-zero elements of $\hat{a}_{\gamma_{j^*},\tau}$ can be used to identify the location of anomalies. If a non-localized basis is chosen, one may use thresholding to determine the anomalous region by $1(\hat{a}_{\gamma_{j^*},\tau} > \omega)$, where ω can be chosen by Ostu's method [43].

4.4 Simulation Study

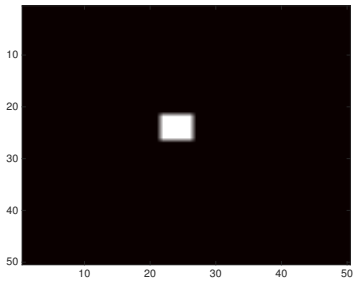
In this section, the performance of the proposed methodology is evaluated by using simulated streams of images with a dynamic functional mean (background). To simulate the functional mean with smooth spatial and temporal structures, we mimic a heat transfer process, in which a 2D temperature map, $M(x, y, t)$, are generated according to the following heat transfer equation [110]:

$$\frac{\partial M}{\partial t} - \alpha(\frac{\partial^2 M}{\partial x^2} + \frac{\partial^2 M}{\partial y^2}) = f$$

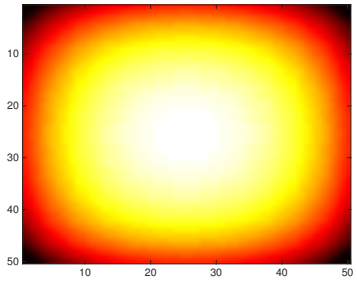
where x, y, $0 \leq x, y \leq 1$ denote pixel locations on an image, α is the thermal diffusivity constant describing how fast a material can conduct thermal energy, and f describes the internal heat generation of the entire surface. In this simulation study, we set $\alpha = 1$. The initial and boundary conditions are set as $M|_{t=0} = 0$ and $M|_{x=0} = M|_{x=1} = M|_{y=0} = M|_{y=1} = 1$, respectively. At each time t, the functional mean $M(x, y, t)$ is recorded at points $x = \frac{i}{m+1}, y = \frac{j}{m+1}$; $i, j = 1, \cdots, m$, which results in an $m \times m$ matrix denoted by $M(t)$. In this study, we consider two types of anomalies; namely, clustered and scattered anomalies. Both types of anomalies are generated based on $S_0 = \delta I(s \in S_A)1(t > t_1)$, in which S_A is the set of anomalous pixels, δ characterizes the intensity difference between anomalies and the functional mean, $1(\cdot)$ is an indicator function, and t_1 is the time of the change. For the scattered case, S_A is a set of 25 pixels randomly selected throughout the



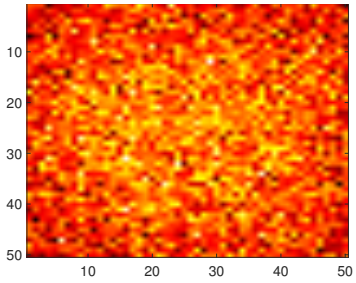
(a) Simulated scattered anomalies after time $t = 200$



(b) Simulated cluster anomalies after time $t = 200$



(c) Simulated functional mean at time $t_1 = 201$



(d) Simulated image coupled with noise and anomalies at time $t_1 = 201$

Figure 4.1: Simulated images with both functional mean and anomalies at time $t = 201$

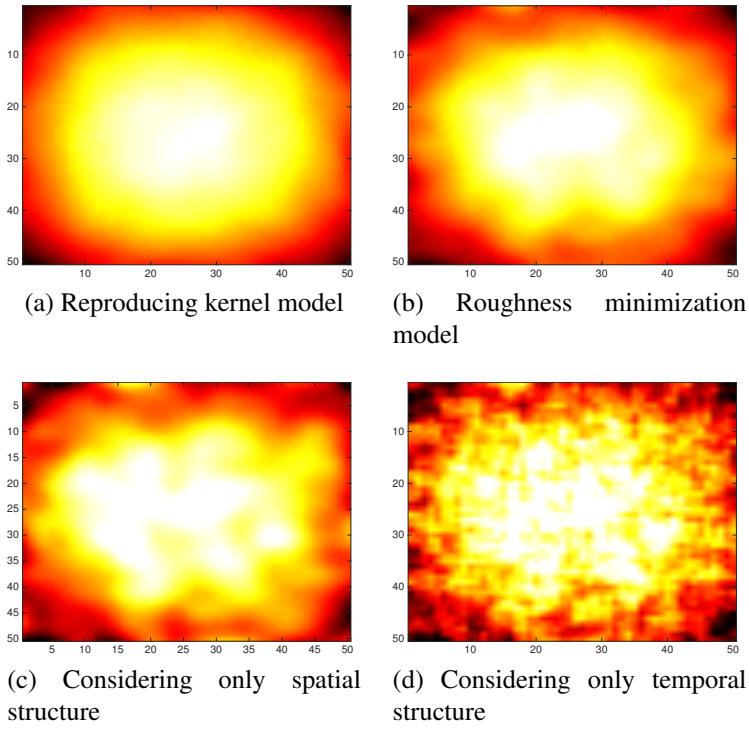
image. For the clustered case, S_A is a randomly generated 5×5 square. Finally, the matrix of random noises, i.e., $E_i \sim NID(0, \sigma^2)$ with $\sigma = 0.1$, are added to the generated image streams. A sample of simulated scattered and square anomalies, the simulated functional mean, and an example of simulated noisy image are shown in Figure 4.1a, (b), (c) and (d), respectively.

To model the spatial structure of each image $M(t)$, we use cubic B-spline basis with 10 knots in both x and y directions. For scattered anomalies, since the size of anomalies is very small and their locations are randomly chosen, an identity matrix can be used as the spatial basis. For the clustered anomalies, however, since anomalies form small continuos regions, a cubic B-spline basis with 30 knots is used in both x and y directions. We also include the results of using an identity basis for the clustered case to study the sensitivity of the proposed method to the choice of bases. We then apply both versions of the ST-SSD model (i.e. kernel and roughness minimization) to the simulated streaming images.

We first begin with evaluating the effectiveness of ST-SSD in estimating the functional mean. The estimated functional mean from a sample of data streams using both reproducing kernel (RK) and roughness minimization (RM) models are shown in Figure 4.2a and (b). The mean square errors (MSE) of the estimated mean are 2.320×10^{-5} and 8.400×10^{-5} for RK and RM, respectively, which indicates a slight advantage of the kernel basis due to its flexibility. Also, in order to show the importance of considering both spatial and temporal structures of data, in Figure 4.2c and (d), we plot the estimated functional mean when only either spatial or temporal structure is modeled. To estimate the functional mean with only spatial structure, we apply SSD on each single image with the same spatial spline basis used in the ST-SSD. To estimate the functional mean considering only the temporal structure, we apply the proposed RM method with the identity matrix as the spatial basis. The MSE of the estimated mean for spatial and temporal models are receptively 2.32×10^{-4} and 3.92×10^{-4}, both larger than that of RK and RM. By comparing Figure 4.2 with Figure 5.3b, it is clear that the estimated functional mean by our proposed ST-SSD is much closer to the true functional mean as it takes both spatial and temporal structures into account.

Next, we compare the performance of our method with a few benchmark methods in the literature. Specifically, we compare the proposed reproducing kernel (designated as RK for the identity spatial basis and as RKcluster for the cubic B-spline basis) and roughness minimization (designated as RM for the identity spatial basis and RMcluster for the cubic B-spline basis) methods with the Hotelling T^2 control chart (designated as 'T2'), Lasso-based control chart proposed by [111] (designated as LASSO) and local CUSUM control chart [49] (designated as 'CUSUM'). It should be noted that none of benchmark methods can remove the temporal trend. Therefore, to have a fair comparison, we use a simple moving average filter with the window size of 5 to remove the temporal trend before applying the benchmark methods. We fix the in-control ARL_0 for all methods to be 200 and compare the out-of-control ARL_1 under different anomaly intensity levels,δ.

The average time of computing the monitoring statistics for a sample is given in Ta-



(a) Reproducing kernel model

(b) Roughness minimization model

(c) Considering only spatial structure

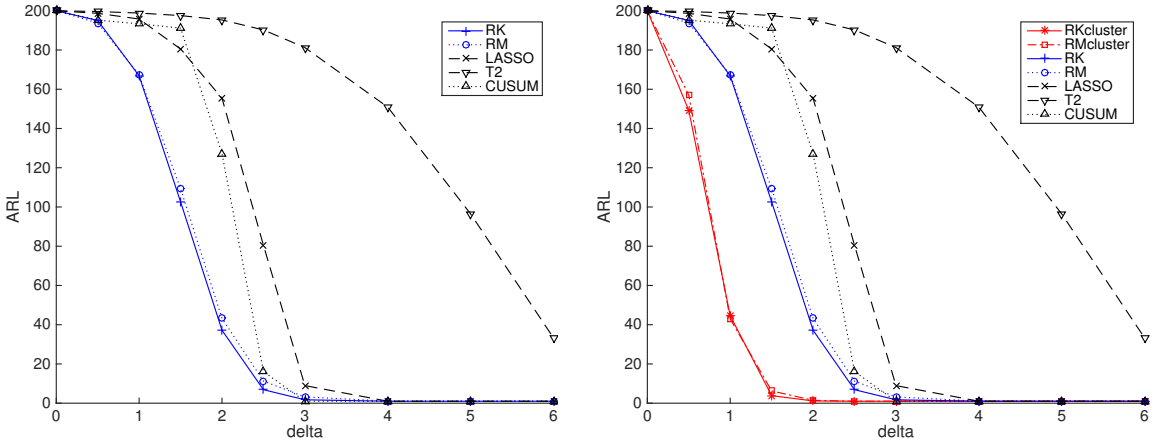
(d) Considering only temporal structure

Figure 4.2: Functional mean estimation results

ble 4.1, and the out-of-control ARL curves of clustered and scattered anomalies obtained from 1000 simulation replications are shown in Figure 4.3a and 4.3b, respectively. In both cases of scattered and clustered anomalies, it is clear that RK and RM models have better detection performance than other benchmark methods. The RK method performs slightly better than RM due to its accuracy in modeling the temporal trend. However, RM is slower in terms of the computation time because of its higher modeling complexity. The reason for the poor performance of the local CUSUM and lasso-based control charts is that they lack the ability to model both the spatial structure and the temporal trend at the same time. Hotelling T^2 control chart performs the worst because it is based on a multivariate hypothesis test, whose power deteriorates as the data dimensions increase, hence, not scalable to HD data streams. Moreover, in the case of clustered anomalies, the proposed RK and RM models with spline basis detect the changes significantly quicker than those with identity basis. For example, in the clustered anomaly case, for a small shift with $\delta = 1$, the ARL

Table 4.1: Computation time of ST-SSD and other benchmark methods

	RK	RM	LASSO	CUSUM	T2
Time	0.13s	0.015s	5.2e-3s	2.0e-4s	1.6e-4s



(a) ARL for scattered defect in different δ

(b) ARL Run Length for clustered defect in different δ

Figure 4.3: Detection power comparison based on ARL

for both RKcluster and RMcluster is around 40, while the ARL for other methods without considering the spatial structure are at least about 4 times larger ($\geq$ 170). This indicates the importance of accurate modeling of the spatial structure in addition to the temporal tend. The ARL of the benchmark methods for such a shift is close to the in-control ARL of 200, indicating that these methods are not capable of detecting small changes. In conclusion, even if the computational time of RK and RM is much larger than LASSO, T2 and CUSUM, it is still small enough to be used for online monitoring. Furthermore, the performance of RK and RM is much better especially in the clustered anomaly case. A video of one simulation run along with the ST-SSD results and the corresponding control chart is given in the online appendix. [1]

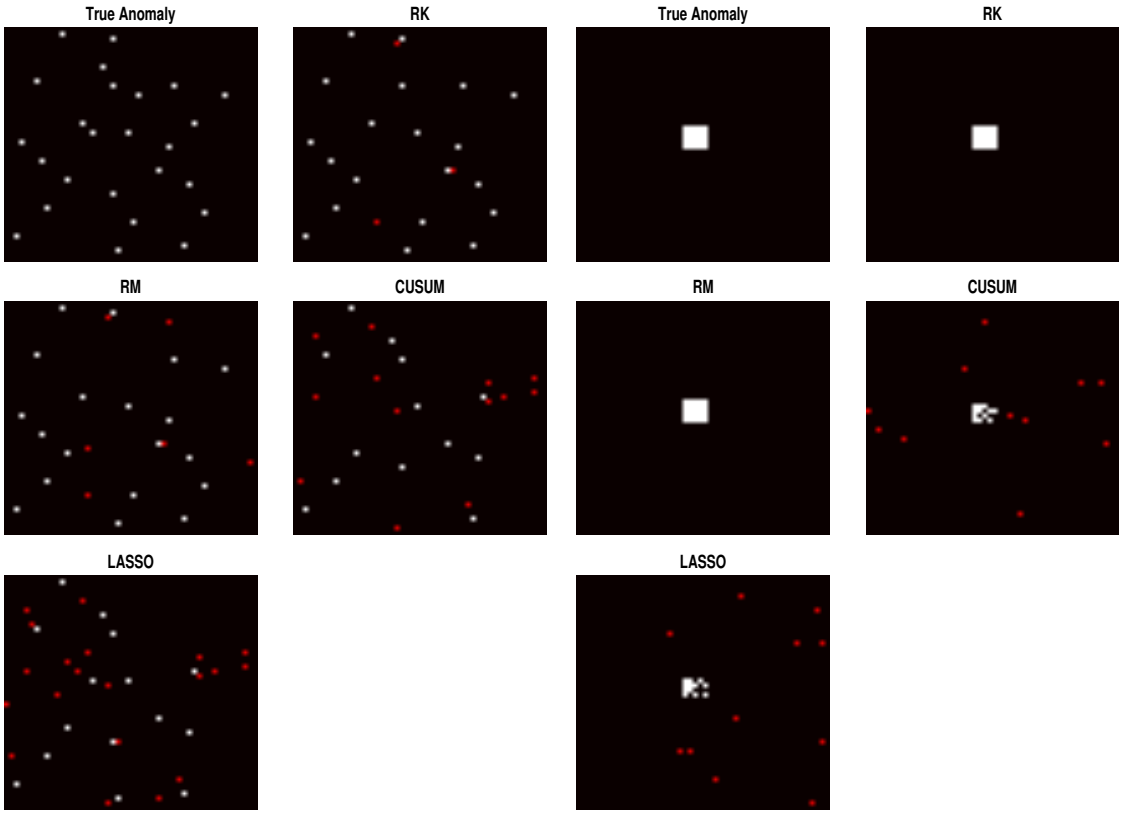
Finally, we evaluate and compare the performance of the diagnosis method with benchmark methods. For this purpose, we compute the following four criteria after a shift is detected: (i) precision, defined as the proportion of detected anomalies that are true anoma-

[1] Online appendix on https://www.dropbox.com/sh/5qf1z8ls5afnpiv/AACRJ3G5lSpxePXFIByXoaqRa?dl=0

Table 4.2: Monitoring and diagnostics result when $\delta = 2$ and $\delta = 3$, (precision, recall and F, the larger the better; ARL, the smaller the better.)

methods	Scattered Anomalies $\delta = 2$				Scattered Anomalies $\delta = 3$			
	precision	recall	F	ARL	precision	recall	F	ARL
RK	0.2357	0.2764	0.2544	37.17	0.6106	0.5500	0.5738	1.73
RM	0.2535	0.2532	0.2533	43.11	0.5851	0.5560	0.5656	1.83
LASSO	0.2553	0.2204	0.2366	155.39	0.5719	0.4892	0.5257	8.96
CUSUM	0.1092	0.1136	0.1114	124.88	0.5187	0.5394	0.5289	1.86
T2	-	-	-	195.32	-	-	-	181.23
methods	Clustered Anomalies $\delta = 2$				Clustered Anomalies $\delta = 3$			
RKcluster	0.8515	0.8596	0.8415	1.11	0.9424	0.9464	0.9444	1.00
RMcluster	0.8490	0.7934	0.8202	1.46	0.9163	0.9474	0.9316	1.00
LASSO	0.2498	0.2160	0.2297	153.88	0.5880	0.4952	0.5333	8.57
CUSUM	0.1100	0.1144	0.1121	121.85	0.5195	0.5402	0.5296	1.95
T2	-	-	-	195.32	-	-	-	181.23

lies; (ii) recall, defined as the proportion of the anomalies that are correctly identified; (iii) F measure, a single criterion that combines the precision and recall by calculating their harmonic mean; and (iv) the corresponding ARL. The average values of these criteria over 1000 simulation replications for $\delta = 2$ and $\delta = 3$ are given in Table 4.2. An example of detected anomalies for both scattered and clustered cases with $\delta = 3$ are also shown in Figure 4.4, in which incorrectly classified points are shown in red. It is clear from this figure and Table 4.2 that the proposed RK and RM models have a much better diagnostics performance than other benchmark methods. This difference is more pronounced in the clustered case where the benchmark methods fail to model the spatial structure of anomalies. For example, for the scattered case, the F measure of both RK and LASSO is around 0.25. However, in the clustered case, this measure is 0.84 for kernel, while lasso's measure remains the same. Moreover, the diagnostics measures of ST-SSD methods (i.e. RK and RM) in the clustered case is much better than the corresponding measures in the scattered case. This is because the spatial structure of defects in the clustered case is well captured by the B-spine basis.



(a) Scattered defects with $\delta = 3$

(b) Clustered defects with $\delta = 3$

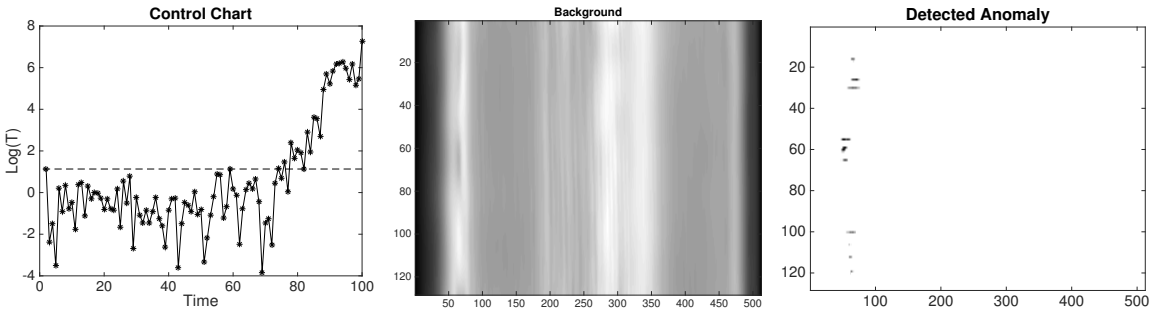
Figure 4.4: Detected anomalies by using different methods (incorrectly identified pixels are shown in red)

4.5 Case study

In this section, the proposed monitoring method is applied to three real datasets collected from a steel rolling process, a solar data observatory, and a stamping process. In the first two cases, we analyze images with a dynamic functional mean and in the third case we study multi-channel profiles with a static functional mean.

4.5.1 On-line Seam Detection in Steel Rolling Process

Rolling is a high-speed deformation process that uses a set of rollers to reduce the cross-section of a long steel bar by applying compressive forces for achieving certain uniform diameters [112]. Surface defects such as seam defects can result in stress concentration on the bulk material that may cause failures when a steel bar is used. Therefore, early detection of anomalies is vital to prevent product damage and to reduce manufacturing costs. Traditionally, due to the high speed of the rolling process (e.g. 225 mile per hour), seam detection has been limited to off-line manual inspection. In recent years, with the development of advanced sensing and imaging technologies, vision sensors have been successfully adopted in rolling processes, collecting high-resolution images of the product surface with a high data acquisition rate. In this case study, a stream of surface images of a rolling bar is used to validate our methodology. We collect a sample of 100 images with the size of 128×512 pixels. The first 50 images are in-control samples with no defects. As an example, one frame of the image stream is shown in Figure 1.4a. An image of a rolling bar is generally smooth in the rolling direction (vertical direction). Moreover, seam defects that have a high contrast against the functional mean (image background) are typically sparse [113], which justifies the use of ST-SSD model for analyzing this data stream. We apply the proposed RM method to monitor rolling process and detect potential defects on the surface. To model the functional mean in y direction, a B-Spline basis with 5 knots is used for B_y and $B_x = I_x$. We also use an identity matrix basis for anomalies in both the x and



(a) Log of testing statistics (b) Functional mean estimation (c) Anomalies estimation

Figure 4.5: Detection results for rolling example at time $t = 97$

y directions, i.e., $B_{ax} = I_{ax}$ and $B_{ay} = I_{ay}$.

Since the dynamic behavior of the functional mean is not intricate, the roughness minimization model described in Section 4.2, is used. The testing statistic in (4.12) is calculated and plotted in Figure 4.5a. The control limit for this case and other examples presented in this section is determined using the in-control data and according to the procedure presented in Section 5.2. Seam defects often occur towards the end of the rolling bar. It is clear from the image stream (see the online appendix), the first defect appears at time $t = 76$, which is the first out-of-control point in the control chart. The computational time is 0.35s per sample, which is sufficiently fast for online monitoring. To illustrate the effectiveness of the diagnosis procedure, the estimated functional mean and detected defects in one out-of-control image recorded at $t = 97$ is shown in Figure 4.5b and 4.5c, respectively. The original image is also shown in Figure 1.4a. As we can see from Figure 4.5, the estimated functional mean (background) is smooth in the y direction and the detected defects are sparse and demonstrate certain repeated patterns suggesting that the roller may be damaged.

4.5.2 Online Monitoring of Solar Activity

In the second example, a stream of solar images are used for monitoring of solar activities and detection of solar flares. A solar flare emits a large number of energetic charged

particles, which may potentially cause the failure of large-scale power-grids. Thus, quick detection of solar flares is important for preventive and corrective actions. The solar temperature slowly changes over time and solar bursts are sparse in both the time and space, which makes process monitoring challenging. Existing detection methods that simply remove the functional mean (background) by subtracting the sample mean are incapable of detecting small transient flares in the dynamic system [114].

This dataset is publicly available online at http://nislab.ee.duke.edu/MOUSSE/index.html. In this dataset, a sequence of images of size 232×292 pixels was captured by satellite. A sample of 300 frames is used in this case study and the first 100 frames are considered as the in-control sample. To detect the solar flare in real-time, the proposed RM monitoring method is applied with the following specification: To model the smooth functional mean (background), B-Spline basis with 50 knots are used as B_x and B_y; to model the sparse anomalies (solar flares), we select the identity matrix for the anomalies in both the x and y directions, i.e., $B_{ax} = I_{ax}$ and $B_{ay} = I_{ay}$. The logarithm of the test statistic obtained from (4.12) is plotted in Figure 4.6. As can be seen form the control charts, three solar flares are detected. The first two solar flares occurred at intervals $[191,\ 194]$ and $[216,\ 237]$, which is compatible with the results reported in [114] and [50]. Additionally, we are able to detect a third small flare at the interval $[257,\ 258]$, which was not detected by the existing two-step approaches (i.e., [50, 114]). Computation time is about 0.12s per frame, which enables online monitoring. Note that although image frames in both case studies have similar number of pixels, the computation time for the analysis of solar images is smaller than that of rolling images. The reason is that the computational complexity for the proposed algorithm is $O(n_x^3 + n_y^3)$, which is in order of 1.2×10^8 and 4×10^7 for rolling and solar images, receptively. This makes the computation time for solar images approximately three times lower.

Furthermore, to find the location of the solar flares in out of control images, the estimated functional mean (the background) and anomalies (solar flares) corresponding to time

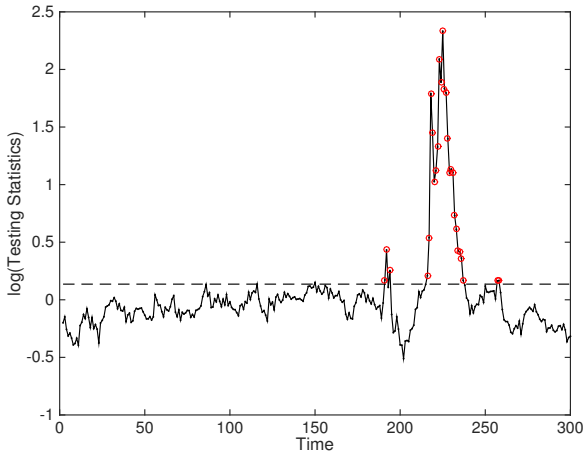


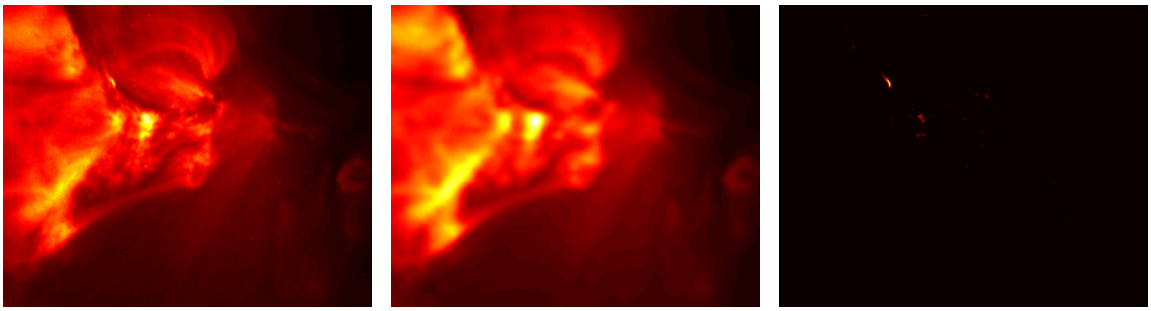
Figure 4.6: Log of testing statistics in solar flare monitoring

$t = 192, 222, 258$ are shown in Figure 4.7. As can be seen form the figure, the proposed method not only is able to detect the changes, but also can identify the location of solar flares in different time frames.

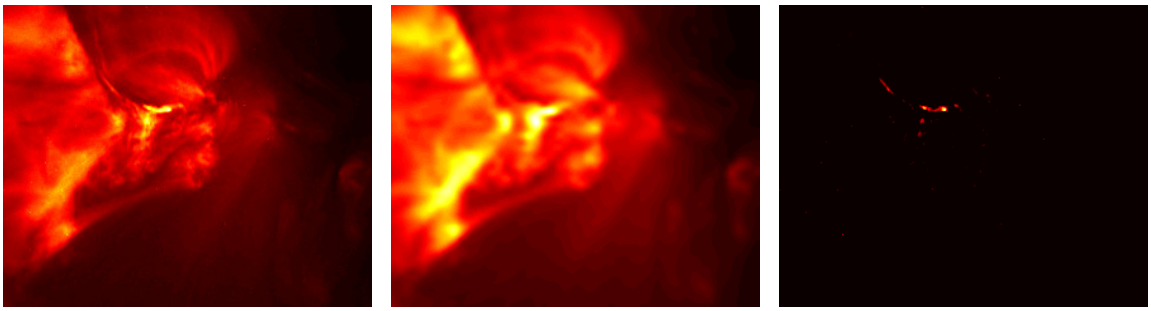
4.5.3 Tonnage Signal Monitoring

We also utilize the proposed methodology to monitor multi-channel tonnage profiles collected in a multi-operation forging process. In this process, four strain gauge sensors, each mounted on one column of the forging machine, measure the exerted tonnage force of the press uprights as shown in Figure 4.8a. This results in a four-channel tonnage profile in each cycle of operation. The dataset used in this case study contains 202 in-control profiles collected under normal production condition and 69 out-of-control profiles in which there is a missing part in the piercing operation die. As pointed out by [1], [115] and [59], a missing part only affects certain segments of the tonnage profile, which implies that the change is sparse. Hence, in this case study, we only focus on the peak area of the tonnage profile, which is mostly affected by a missing part. The length of the peak profiles for each channel is 569. Examples of peak profiles for both normal and faulty conditions are shown in Figure 1.4c.

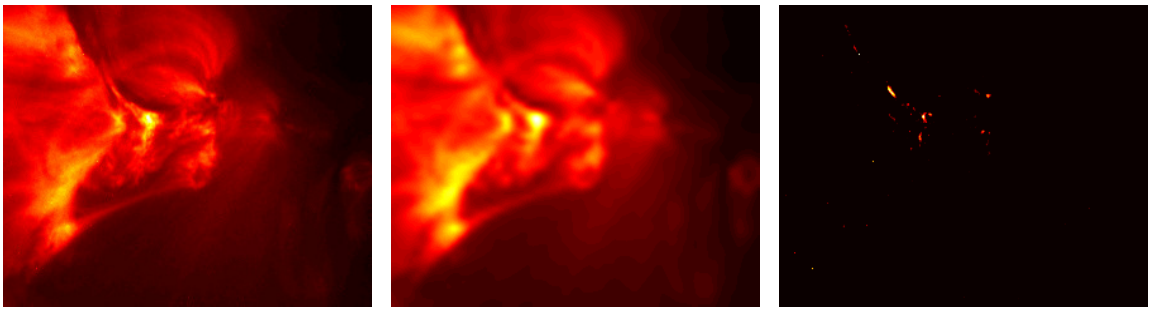
Since the signal mean is static, the proposed static model is applied. However, to model



(a) First solar flare at t=192 (b) Mean estimation at t=192 (c) Anomalies estimation at t=192

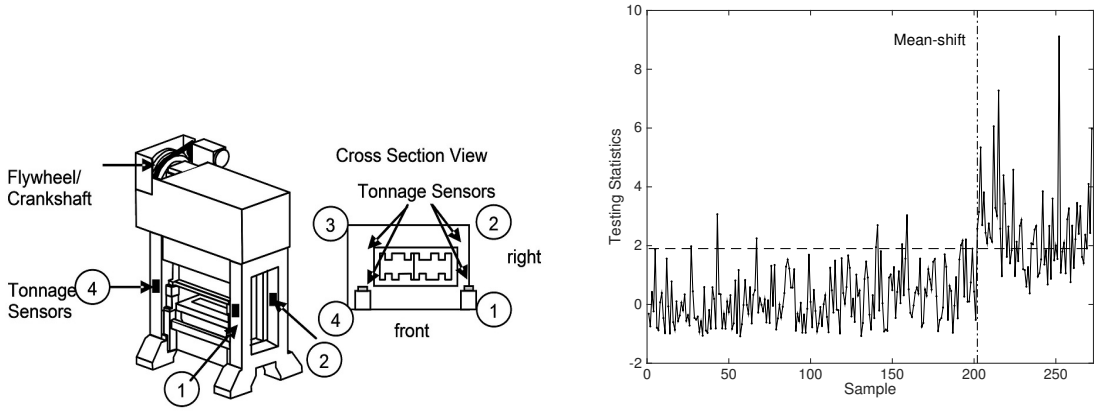


(d) Second solar flare at t=222 (e) Mean estimation at t=222 (f) Anomalies estimation at t=222



(g) Third solar flare at t=258 (h) Mean estimation at t=258 (i) Anomalies estimation at t=258

Figure 4.7: Detection results in three solar frames at time $t = 192, 222, 258$



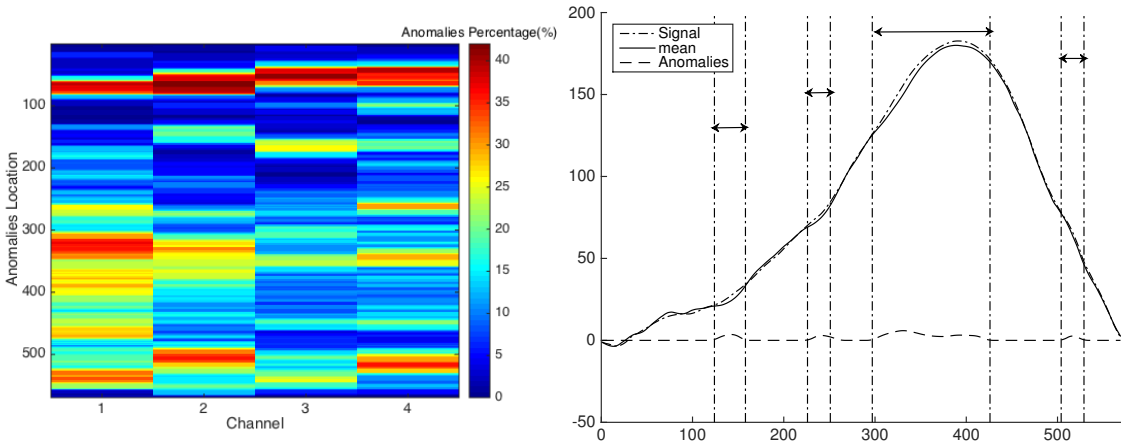
(a) A Stamping press and tonnage sensor location

(b) Tonnage Signal Monitoring Result

Figure 4.8: Monitoring forging process using multi-channel tonnage signal

the spatial structure of the profile mean and anomalies, cubic B-spline bases with 10 and 90 knots are used, respectively. We use the sequence of in-control profiles to estimate the control limit. Out of 202 samples collected under the normal operations, 9 samples are specified as out-of-control. After removing these outlier samples and recalculating the control limit, the proposed monitoring method is applied to the sequence of faulty profiles and the resulting control chart is shown in Figure 4.8b. As shown in the figure, there is a clear change in the mean of the monitoring statistic, indicating that the monitoring method can detect the profile changes caused by missing parts. Overall 44 out of 69 faulty samples are beyond the control limit, which is roughly equivalent to the out-of-control ARL of 1.5. The computational time on average is $0.25s$ per sample.

Moreover, we use all out-of-control samples to perform diagnosis analysis. The percentage of identified anomalies by our diagnosis method across different channels and segments are shown in a colormap in Figure 4.9a. Warmer colors imply that more out-of-control samples contain anomalies in the corresponding channel segment. As can be seen in Figure 4.9a, anomalies mostly occur in the segment $[44, 88]$, segment $[319, 346]$ and segment $[497, 535]$ and mostly in Channel 1. This is because Sensor 1 is mounted on the front side of the forging machine where the die with missing parts is located. Figure



(a) Diagnostics of the tonnage signal

(b) Tonnage Signal Anomalies location

Figure 4.9: Tonnage Signal Diagnostics

4.9b shows one example of faulty profile recorded by Sensor 1 along with the profile mean and the identified anomalous segment. As can be seen from the figure, the main difference between the signal and the profile mean is picked up by the the diagnosis procedure. These findings are consistent with those in [1, 59].

4.6 Conclusion

Online monitoring of high-dimensional streaming data with complex spatio-temporal structure is very important in various manufacturing and service applications. In this chapter, we proposed a novel methodology for real-time monitoring of HD data streams. In our methodology, we first developed ST-SSD that effectively decomposes a data stream into a smooth functional mean and sparse anomalies by considering the difference in the spatio-temporal structures of the functional mean and anomalies. Similar to SSD, we formulated ST-SSD in the form of high-dimensional regression augmented with penalty terms to encourage both the smoothness of the spatio-temporal functional mean and the sparsity of anomalies. To effectively solve this large-scale convex optimization problem, we used APG methods and developed efficient iterative algorithms that have closed-form solutions in each iteration. This method can be applied to identify anomalies and the functional mean

for a fixed number of samples, which can only be applied in offline phase-I monitoring. To handle challenges of the increasing number of observations in online monitoring, reproducing kernel and roughness minimization models were developed as two temporal modeling methods that provide a recursive estimation scheme for ST-SSD. This enables real-time implementation of ST-SSD. Then, a sequential likelihood-ratio-test-based control chart was proposed for monitoring. In the simulation study, we showed that the proposed methods outperforms existing process monitoring approaches that fail to effectively model both the spatial structure and temporal trend. Finally, the proposed method was applied to three real case studies including steel rolling, solar activity, and tonnage signal monitoring. The results from all case studies demonstrated the capability of the proposed methods in identifying not only the time of process changes, but also the location of detected anomalies.

chapter

CHAPTER 5
AN ADAPTIVE FRAMEWORK FOR ONLINE SENSING AND ANOMALY DETECTION

In this chapter, we develop a novel framework named Adaptive Kernelized Maximum-Minimum Distance $(\mathrm{AKM^2D})$ to speed up the inspection and anomaly detection process through an intelligent sequential sampling scheme integrated with fast estimation and detection. The proposed method balances the sampling efforts between the space filling sampling (exploration) and focused sampling near the anomalous region (exploitation). The proposed methodology is validated by conducting simulations and a case study of anomaly detection in composite sheets using a guided wave test.

The remainder of the chapter is organized as follows. Section 5.1 provides an overview of the proposed methodology. In Section 5.2, we propose the new adaptive sampling/sensing framework $\mathrm{AKM^2D}$. Section 5.3 elaborates mean estimation and anomaly detection algorithms. In Sections 5.4 and 5.5, simulated data and a case study of anomaly detection in composite laminates are used to evaluate the performance of the proposed methodology. Finally, we conclude the chapter with a short discussion in Section 5.6.

5.1 Methodology Overview

We first briefly review the overall methodology proposed in this chapter, which includes two main components: an adaptive sampling framework and procedures for estimating the functional mean and anomalies. For illustration purposes, we use the $2D$ sampling space $[0,1]^2$ in this chapter. We further constrain the samples to be on a $2D$ fine grid defined as $\mathcal{G}_m = \{(\frac{i}{m}, \frac{j}{m}) | i,j = 1, \cdots, m\}$, where m can be specified by the resolution capability of the sensing device. It should be noted that the proposed methodology can be easily extended to a higher dimensional space or continuous space.

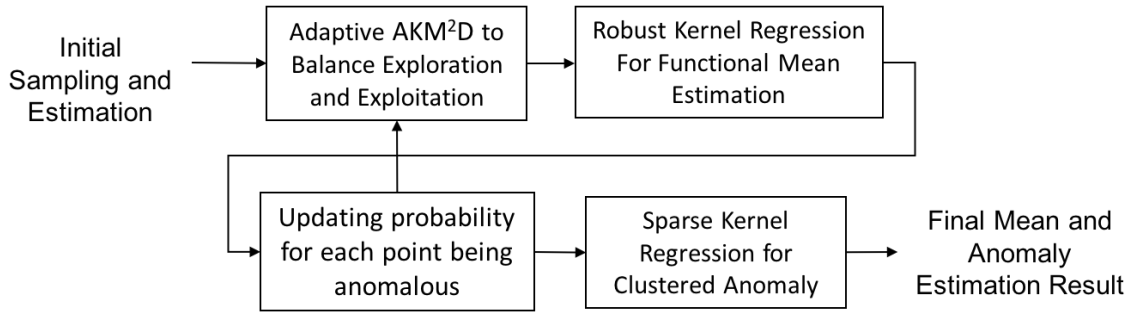


Figure 5.1: Procedure of the proposed sampling algorithm

The proposed methodology, illustrated in Figure 5.1, is summarized as follows: First, n_{init} initial points are sampled using a space-filling design (e.g. max-min distance, [116]) to explore the entire sampling space. Then, based on the outcome of the initial points, subsequent points are chosen by using $\mathrm{AKM^2D}$ to balance between the space-filling sampling (exploration) and the focused sampling near the anomalous region (exploitation). After $\mathrm{AKM^2D}$ chooses the location of a new sample, the functional mean is estimated (updated) via robust kernel regression. After certain number of sampled points, if the functional mean estimate does not deviate much from the estimate obtained in the previous iteration, the functional mean estimation step can be skipped to reduce the computational time. Also, in this step, the probability that a point in the sample space is anomalous is updated, which is an input for $\mathrm{AKM^2D}$ in the next iteration. Next, clustered anomalous regions are estimated (updated) via the proposed sparse kernel regression. Finally, this procedure is repeated until the desired sampling resolution is reached.

In developing the proposed sampling methodology, we make the following assumptions: we assume that sparse anomalies are in the form of clusters. Also, for estimating the functional mean and anomalous regions, it is assumed that the functional mean is smooth and anomalies have different intensity values from the functional mean. It should be noted that the proposed $\mathrm{AKM^2D}$ framework is general and does not require the smoothness assumption.

5.2 Adaptive Kernelized Maximum Minimum-Distance ($\mathrm{AKM^2D}$) Sensing

5.2.1 Formulation and Algorithm

In this section, we present our new adaptive sensing framework, $\mathrm{AKM^2D}$, that helps sequentially choose the location of samples. Suppose n sampled points located at $\mathcal{M}_n = \{r_k = (x_k, y_k) \in \mathcal{G}_m | k = 1, \cdots, n\}$ are observed in an iteration. Let $p_a(r_k)$ denote the known probability that the point r_k in this set is anomalous (the detailed procedure for estimating $p_a(r_k)$ will be discussed in Section 4.) To find the next sampled point r_{n+1}, we propose the following criterion:

$$r_{n+1} = \arg\max_r g_n(r) = \psi_n(r)(f_n(r))^\lambda, \tag{5.1}$$

where $\psi_n(r)$ is the estimated distribution of anomalies. Therefore, maximizing $\psi_n(r)$ can encourage the focused sampling (exploitation) meaning that the next sampled point r_{n+1} continues searching in anomalous regions. $f_n(r)$ is the regularization term to prevent sampled points being too close to each other. In the other word, $f_n(r)$ encourage the exploration of entire sampling space for undiscovered anomalies (space-filling property). In this chapter, we define $\psi_n(r)$ as a mixture distribution of gaussian distributions centered at each anomalous point observed, and a uniform distribution for the entire sampling space to account for unobserved anomalies. That is, $\psi_n(r) = (\sum_{k=1}^n p_a(r_k) K_h(r, r_k) + u)$ where $K_h(r, r_k) = \frac{1}{(\sqrt{2\pi}h)^2} \exp(-\frac{\|r-r_k\|^2}{2h^2})$ is the 2D-gaussian kernel centered at point r_k used to model the clustered structure of the anomalies. $p_a(r_k)$ and u are respectively the mixture weights for the gaussian distribution $K_h(r, r_k)$ and the uniform distribution. Note that $p_a(r_k)$ is also the probability that the sampled point r_k is anomalous. The normalization weight $\frac{1}{\sum_{k=1}^n p_a(r_k)+u}$ is neglected since it is constant and independent of r. Furthermore, we define $f_n(r)$ by $f_n(r) := \min_{r_k \in \mathcal{M}_n} \|r - r_k\|$ to encourage the space-filling property. For a special case $\psi_n(r) = 1$, Equation (5.1) becomes $r_{n+1} = \arg\max_r \min_{k=1,\cdots,n} \|r - r_k\|$

Algorithm 7: AKMMD

initialize
  Initial n_{init} sampling based on max-min distance design
end
for $n = n_{init}, \cdots, n_{max}$ **do**
  Update $\psi_n(r)$ based on $\Psi_n = K_x^T P_A K_y + u1_{m\times m}$
  Update $f_n(r) = \min(f_{n-1}(r), \|r - r_n\|)$ for $r \in \mathcal{G}_m$
  $r_{n+1} = \text{argmax}_{r\in\mathcal{G}_m}\psi_n(r)(f_n(r))^\lambda$
end

which is equivalent to a greedy approach to solve the maximum minimum-distance design proposed by [116]. By plugging in $\psi_n(r)$ and $f_n(r)$, the sampling criterion given in (5.1) can be rewritten as

$$r_{n+1} = \text{argmax}_r \left\{ (\sum_{k=1}^{n} p_a(r_k)K_h(r, r_k) + u) \min_{k=1,\cdots,n} \|r - r_k\|^\lambda \right\}. \tag{5.2}$$

To efficiently solve (5.2) on $r \in \mathcal{G}_m$, we compute $\psi_n(r)$ by the tensor product of two 1D-gaussian kernel. That is, $\Psi_n = K_x^T P_A K_y + u1_{m\times m}$, where $K_{x,ij} = K_{y,ij} = \frac{1}{(\sqrt{2\pi}h)^2}\exp(-\frac{\|i-j\|^2}{2h^2m^2})$, and $P_{A,ij} = p_a(\frac{i}{m}, \frac{j}{m})1((\frac{i}{m}, \frac{j}{m}) \in \mathcal{M}_n)$ are the ij component of the matrix K_x, K_y and P_A, respectively. $1(x)$ is an indicator function defined as $1(x) = \begin{cases} 1 & x \text{ is true} \\ 0 & x \text{ is false} \end{cases}$, and $1_{m\times m}$ is an m by m matrix of 1s. It is straightforward to show that $f_n(r), r \in \mathcal{G}_m$ can be updated recursively by $f_n(r) = \min(f_{n-1}(r), \|r - r_n\|), r \in \mathcal{G}_m$. Both the space and time complexity of this recursive update is $O(m^2)$, where m is the grid size in each dimension. Therefore, (5.2) can be efficiently and recursively solved by Algorithm 7.

5.2.2 AKM^{2}D Sampling Properties

In this section, we study the properties of the proposed AKM^{2}D. Let $\mathcal{R}_i$ denote the neighborhood of a point r_i defined by $\mathcal{R}_i = \{r | \|r - r_i\| \leq \|r - r_k\|, \forall k = 1, \cdots, n\}$. We first investigate the behavior of the sampling criterion $g(r)$ in the neighborhood of an anomalous point r_a, i.e., $\mathcal{R}_a$. (see Figure 5.2). It is easy to show that Equation (5.2) for $r \in \mathcal{R}_a$

can be decomposed into two terms: $g(r) = g_a(r) + g_{-a}(r)$, where $g_a(r) = (K_h(r, r_a) + u)\|r - r_a\|^\lambda$, $g_{-a}(r) = \left(\sum_{k\neq a}^{n} p(r_k)K_h(r, r_k)\right)\|r - r_a\|^\lambda$. The second term, $g_{-a}(r)$, is often negligible in the neighborhood of r_a especially when $\|r_k - r_a\| \gg h, \forall k \neq a$. For simplicity, we assume r_a is the only detected anomalous point with $p_a > 0$.

Proposition 9. *The local maximum of $g_a(r), r \in \mathcal{R}_a$ is attained at $\|r - r_a\| = d_a^* = h\sqrt{\lambda - 2W(-\frac{\pi h^2 \lambda u}{p_a}\exp(\frac{\lambda}{2}))}$ if $\{r : \|r - r_a\| = d_a^*\} \in \mathcal{R}_a$. W is the Lambert W-function defined as $W(z) = \{w|z = w\exp(w)\}$.*

Proof is given in Appendix A.

Proposition 9 guarantees that $g_a(r)$ in the neighborhood of r_a will generate a local maximum ring with radius d_a^* (as shown in Figure 5.2), which encourages the next sampled point to be chosen near the potential anomalous point r_a (exploitation), but with the distance of d_a^* to avoid over-exploitation. Proposition 9 only guarantees the local optimality. However, the next sampled point is selected on the local maximum ring only if it is the global maximum of $g(r)$. To study this and show how criterion (5.2) is able to balance sampled points between exploration and exploitation, we give the following necessary condition under which the algorithm selects r_a^* .

Proposition 10. *Let d^* denote the current sampling Max-Min Distance (MMD) defined as the maximum distance of each point in the entire sampling space with its closest sampled point, i.e., $d^* := \max_r \min_{r_k\in\mathcal{M}_n} \|r - r_k\|$ and suppose r_a is the only sampled point with $p_a > 0$. If there exists a constant c such that $\|r_k - r_a\| \geq \max(2c\sqrt{2h^2\ln(\frac{p_a}{2\pi h^2 u})}, 2d_a^*)$, then $\|r - r_a\| = d_a^* = h\sqrt{\lambda - 2W(-\frac{\pi h^2\lambda u}{p_a}\exp(\frac{\lambda}{2}))}$ is the global maximum of* (5.2) *if*

$$d^* < \tilde{d^*} := \left(\frac{1}{(1 + \exp(-c^2)} \times \frac{(d_a^*)^2}{2((d_a^*)^2 - \lambda h^2)}\right)^{\frac{1}{\lambda}} d_a^*. \tag{5.3}$$

Proof is given in Appendix B.

Proposition 10 shows that the proposed algorithm first samples the entire space up to a certain resolution $\tilde{d^*}$ and then, starts focused sampling. This ensures that the proposed

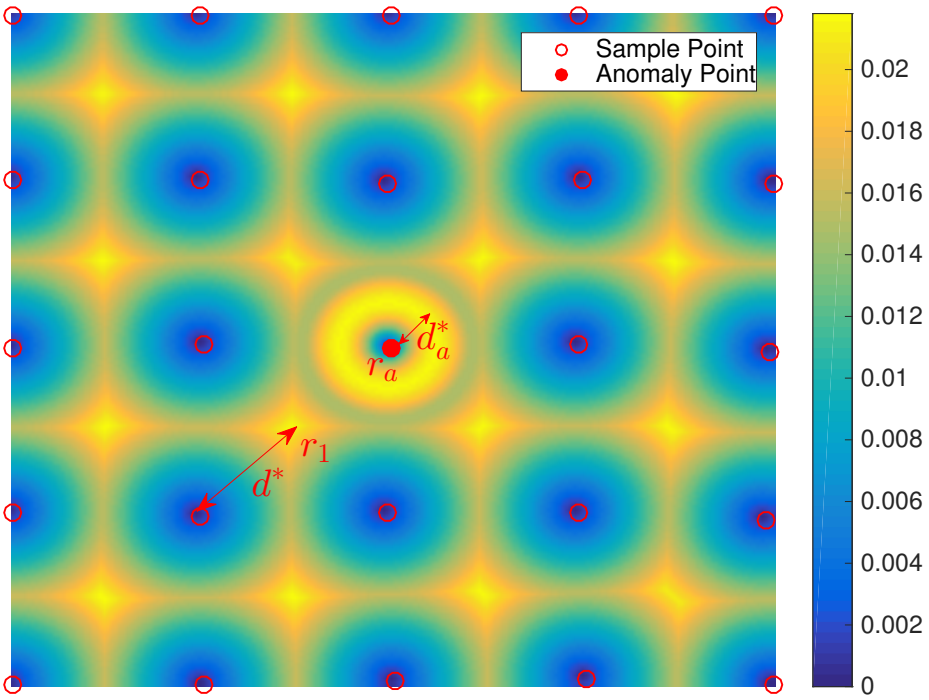


Figure 5.2: Behavior of $g(r)$ with the center point as anomaly point

method does not miss any anomaly with radius greater than $\tilde{d}^*$. Furthermore, this proposition can be used for choosing the tuning parameters, which will be discussed in the next section.

To illustrate the implication of this proposition, we plot the behavior of $g(r)$ in Figure 5.2. The center point in this figure is an anomalous point (the point indicated by r_a), which generates a local optimal ring with radius d_a^*. It will be global optimum if this optimal value is larger than the other local maximum in the center of the potential sampled points (the point indicated by r_1) as shown in Figure 5.2. Proposition 10 shows that if (5.3) holds, the algorithm will select a point on the local maximum ring centered at r_a as the global optimum and hence as the next sampled point.

5.2.3 Tuning Parameter Selection

In this section, we discuss how to select tuning parameters λ, h and u. First, based on our numerical experiments in the simulation study, we suggest the kernel bandwidth h is

selected approximately as the $1/5$ of the anomaly size desired to be detected. To avoid over-exploitation for subsequent sampled points λ should be large. We found $\lambda > 5$ works reasonably well in practice. Furthermore, according to proposition 10, the desired sampling MMD, $\tilde{d}^* = (\frac{(d_a^*)^2}{2((d_a^*)^2 - \lambda h^2)})^{\frac{1}{\lambda}} d_a^*$, and the focused sampling radius d_a^*, can also be used as a guideline to select the tuning parameters. Note that when computing $\tilde{d}^*$, we ignore the $(\frac{1}{1+\exp(-c^2)})^{\frac{1}{\lambda}}$ since it is close to 1 when $c > 3$ and $\lambda > 5$. For example, $c = 3$, $\lambda = 5$, $(\frac{1}{1+\exp(-c^2)})^{\frac{1}{\lambda}} = 0.99998$.

5.3 Mean and Anomaly Estimation Using Sparse Samples

In the previous section, we proposed a general adaptive sampling strategy and discussed its properties. Here, we propose methods for estimating the mean function as well as anomalous regions using the sparse measurements obtained by $\mathrm{AKM^2D}$. Specifically, we present a robust kernel regression algorithm for functional mean estimation and a sparse kernel regression algorithm for anomaly estimation.

5.3.1 Robust Kernel Regression for Functional Mean Estimation

Let z_k denote the recorded measurement at point $r_k = (x_k, y_k)$ and $z = (z_1, \cdots, z_k, \cdots, z_n)$ be the vector of measurements for all n sampled points. To model the smooth functional mean μ in the presence of anomalies, Reproducing Kernel Hilbert Space (RKHS) is utilized. From the representer theorem [106], it is known that every function in an RKHS can be written as a linear combination of kernel functions evaluated at sampled points. If anomalies did not exist, kernel regression could be used for estimating the functional mean. However, since anomalies have a different functional structure from the mean, they behave as outliers when estimating the functional mean. Therefore, we utilize robust kernel regression to alleviate the effect of anomalies on mean estimation. To estimate the functional mean μ, we minimize

$$\sum_{k=1}^{n} \rho(z_k - \mu_k) + \lambda \|\mu\|_H, \tag{5.4}$$

in which $\rho(x)$ is the Huber loss function, defined by $\rho(x) = \begin{cases} x^2 & |x| \leq \frac{\gamma}{2} \\ \gamma|x| - \frac{\gamma^2}{4} & |x| > \frac{\gamma}{2} \end{cases}$, and $\lambda\|\mu\|_H$ is the Hilbert norm penalty, which controls the smoothness of the functional mean. The Robust kernel regression can be solved efficiently via an iterative soft-thresholding function [117]. See Appendix C for the detailed derivation and optimization algorithm. The functional mean μ is almost the same after sensing enough sampled points. Therefore, to speed up the algorithm, we stop updating μ when the estimation difference after adding a new sampled point is smaller than a certain threshold. After estimating the functional mean μ_k, the residuals can be computed by $\hat{e} = [\hat{e}_k] = [z_k - \hat{\mu}_k]$.

5.3.2 Updating Probability $p_a(r_k)$

We conduct a hypothesis test on the residual $\hat{e}_k$ to test whether there exist anomalies in the specimen at the location r_k. The null hypothesis is $H_0 : \mu_{e_k} = 0$, implying no anomalies exist. The p-value of this test can be used to update the probability of the sampled point r_k being anomalous. That is, $p_a(r_k) = P(|e_k| > |\hat{e}_k||e_k \sim N(0, \hat{s}^2)) = 1 - 2P(e_k > \hat{e}_k) = 2\Phi(\frac{|\hat{e}_k|}{\hat{s}}) - 1$, where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution, $\hat{s}$ is the standard deviation of the noise e, which can be estimated by the median absolute deviation under the normality assumption as $\hat{s} = \text{median}\{|\hat{e}|\}/0.6745$. $p_a(r_k)$ is used as an input to AKM^2D as discussed earlier. Moreover, the selection of γ can be determined based on a specified false positive rate, α_0, associated with the hypothesis test. If no anomalies exist (H_0 is true), the false positive rate can be computed by $P(|e_k| > \frac{\gamma}{2}|e_k \sim N(0, \hat{s}^2)) = 2(1-\Phi(\frac{\gamma}{2\hat{s}})) = \alpha_0$. Consequently, γ can be selected by $\hat{\gamma} = 2\hat{s}\Phi^{-1}(1-\frac{\alpha_0}{2})$. See Appendix C for the reason as to why $\frac{\gamma}{2}$ is a good threshold to determine whether a point is anomalous.

5.3.3 Sparse Kernel Regression for Clustered Anomaly Estimation

In this subsection, we estimate the size, shape, and boundary of anomalous regions. Specifically, we model the spatial structure of clustered anomalies by a gaussian kernel K_a through optimizing

$$\arg\min_{\theta_a} \|\hat{e} - K_a\theta_a\|^2 + \gamma_a|\theta_a|_1. \tag{5.5}$$

Problem (5.5) can be solved efficiently by existing L1 solvers such as the accelerated proximal gradient (APG) method used in [72]. The APG algorithm for solving Problem (5.5) is given in Algorithm 8. For the tuning parameter γ_a, as it has been pointed out by [72], Generalized Cross Validation (GCV) usually tends to select more points, leading to a larger false positive rate. Therefore, instead of using GCV, we choose γ_a based on a specified false positive rate α. Since there is no closed-form solution for Problem (5.5) with general K_a, Monte Carlo simulations can be used to select γ_a as follows: generate white noise from $e \sim NID(0, \hat{s}^2)$, where $\hat{s}$ is the standard deviation of the noise e. Select γ_a such that $\alpha \times 100\%$ of $\hat{a} = K_a\hat{\theta}_a$ are non-zero. Note that since K_a changes overtime, γ_a should be recomputed whenever a new point is measured, which is time-consuming. Therefore, an approximate procedure for tuning parameter selection is proposed. When K_a is orthogonal, θ_a has a closed-form solution computed by $\hat{\theta}_a = S_{\frac{\gamma_a}{2}}(K_a^T\hat{e})$, or equivalently, $\hat{\theta}_{ai} = S_{\frac{\gamma_\alpha}{2}}(\sum_j K_a(r_j, r_i)\hat{e}_j)$. When K_a is close to orthogonal, the soft-thresholding function gives a reasonable approximate solution. The false positive rate can then be computed by $\alpha = P(\hat{\theta}_{ai} \neq 0) = 2P(|z| > \frac{\gamma_a}{2}|z \sim N(0, l^2\hat{s}^2) = 2\Phi(1 - \frac{\gamma_a}{2l\hat{s}})$, where $l^2 = \sum_j K_a(r_j, r_i)^2$. Therefore, γ_a can be approximated by $\gamma_a = 2l\hat{s}\Phi^{-1}(1 - \frac{\alpha}{2})$.

To determine the anomalous regions, since the gaussian kernel is not localized, we threshold the solution to (5.5) using a small threshold w to ensure noises are not detected. Consequently, anomalous regions are estimated by $1(\hat{a} > w)$, where $1(x)$ is an indicator function. In our study, we select $w = 0.005\hat{s}$. Furthermore, as the number of points in anomalous regions increases, the corresponding kernel size should decrease accordingly.

Algorithm 8: APG algorithm for sparse kernel estimation of anomalies

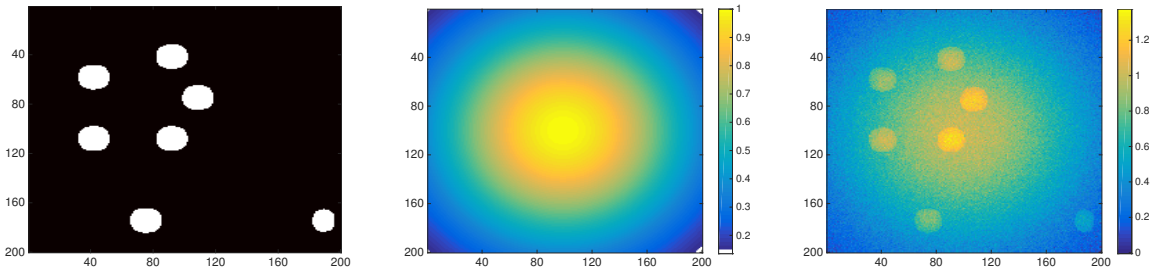
initialize
  Choose a basis for the background as B
  $\theta_a^{(0)} = 0$
end
while $|\theta_a{}^{(k-1)} - \theta_a^{(k)}| > \epsilon$ **do**
  Update $\theta_a^{(k+1)}$ by $\theta_a^{(k+1)} = S_{\frac{\gamma}{2}}(x^{(k)} + K_a^T(e - K_a x^{(k)}))$
  Update $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
  Update $x^{(k+1)} = \theta_a^{(k)} + \frac{t_k - 1}{t_{k+1}}(\theta_a^{(k)} - \theta_a^{(k-1)})$
end

Therefore, we update the bandwidth of kernel K_a (i.e. h_a) proportionally to the sampling resolution in anomalous regions. That is, $h_a = c_h \max_{r \in \hat{a}} \min_{r_k} \|r - r_k\|$. From the simulation study, we found $c_h = 0.2$ works reasonably well.

5.4 Simulation Study

To evaluate the performance of the proposed methodology, we simulate 200×200 images with a smooth functional mean denoted by matrix M whose elements are obtained by evaluating $M(x, y) = \exp(-\frac{(x^2+y^2)}{4})$ at points $x = \frac{i}{201}, y = \frac{j}{201}; i, j = 1, \cdots, 200$. In this study, 7 anomaly clusters are generated by $A = B_s A_s B_s^T$, in which B_s is a cubic B-spline basis with 13 knots, and A_s is a 13 by 13 sparse matrix with 7 randomly selected non-zero entries denoted by S_A. The elements of A_s are defined by $A_s(i, j) = \delta \cdot 1(a_{ij} \in S_A)$, where $\delta = 0.3$ characterizes the intensity difference between anomalies and the functional mean. Random noises E are generated from $E \sim NID(0, \sigma^2)$ with $\sigma = 0.05$. Finally, the set of 200×200 simulated images, Y, is generated by adding the anomalies and random noises to the functional mean, i.e., $Y = M + A + E$. A sample of simulated functional mean, anomalies, and a noisy image with anomalies are shown in Figure 5.3. The goal of this simulation study is to accurately estimate anomalous regions with the least number of sampled points.

We compare our proposed adaptive sampling framework, $\mathrm{AKM^2D}$, with the random



(a) Simulated anomalies (b) Simulated functional mean (c) Simulated images coupled with noise and anomalies

Figure 5.3: Simulated images with both functional mean and anomalies

sampling method (designated by "Random") and multi-resolution grid sampling (designated by "Grid"). In the Random sampling method, the sampled points are selected purely at random. In the Grid sampling, the sampled points are first selected on a 15×15 coarse grid. If $p_a > 0.5$, a finer grid with a five-times-higher resolution is then used to sample within the coarse grid containing anomalous points. We apply the proposed estimation method to the sampled points obtained by both $\mathrm{AKM^2D}$ and the benchmarks to estimate the anomalous regions. In this way, the difference in anomaly detection performance can only be attributed to the sampling strategy.

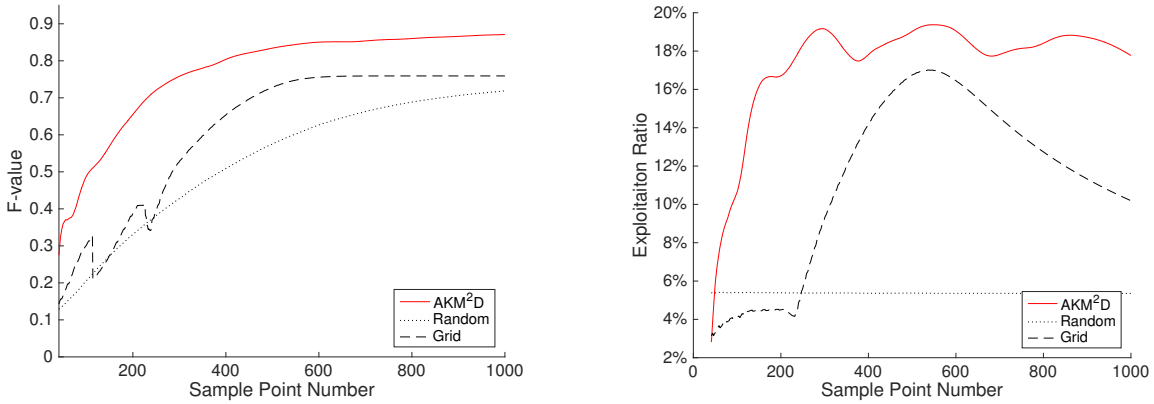
To compare the performance of different sampling methods, the average value of the following criteria are computed over 5000 simulation replications: Precision, defined as the the percentage of detected anomalies by the algorithm that are indeed anomalous; Recall, defined as the percentage of the true anomalous regions detected by the algorithm; F-measure, defined as the harmonic mean of precision and recall; Exploitation Ratio (ER), defined as the percentage ratio of sampled points in the true anomalous regions to total number of sampled points; Anomaly Max-Min Distance (AMMD), defined as the maximum distance of points in the true anomalous region to the nearest sampled point; Max-Min Distance (MMD), defined as the maximum distance of points in the entire sampling space to the nearest sampled point; and the computational time of the sampling procedure for each sampled point. These average values are reported in Table 5.1. Form the table, it

is clear that the proposed $\mathrm{AKM}^2\mathrm{D}$ overall outperforms other benchmark methods. For example, with 250 sampled points, the recall of $\mathrm{AKM}^2\mathrm{D}$ is about 78% indicating that 78% of the anomalous regions have been detected by $\mathrm{AKM}^2\mathrm{D}$ with only 250 points. This is much higher than the recall of benchmarks that is only about 27%. Although benchmark methods have slightly higher precision, the overall classification accuracy, measured by F is in favor of $\mathrm{AKM}^2\mathrm{D}$. The F-measure of $\mathrm{AKM}^2\mathrm{D}$ is around 0.70, while it is around 0.40 for Random and Grid. MMD and AMMD values of the $\mathrm{AKM}^2\mathrm{D}$ are also much smaller than those of Random and Grid, which indicates the proposed $\mathrm{AKM}^2\mathrm{D}$ achieves better exploration of the entire sampling space and better focused sampling near the anomalous regions. Similarly, the ER of $\mathrm{AKM}^2\mathrm{D}$ with 250 sampled points method is around 18%, 3.6 times larger than that of Random and Grid (around 5%). This implies that the proposed method is able to quickly locate anomalous regions and sample about 3.6 times more points in those regions than benchmark methods. Note that the area of anomalous regions covers about 5.8% of the entire sampling space. However, $\mathrm{AKM}^2\mathrm{D}$ with around 0.6% of the full sampled points (250 sampled points out of 200×200), is able to detect at least 78% of the true anomalous regions. If we increase the number of sampled points to 400, this number increases to 88%, whereas for Grid and Random it is around 64% and 40%, respectively. The main reason for the poor performance of Grid is that it lacks the ability of quickly focusing on the discovered anomalous regions. Moreover, the fine sampling grid is rigid, and hence it is not flexible to detect arbitrarily shaped anomalies. Random performs the worst since it does not incorporate any information of detected anomalies. Although $\mathrm{AKM}^2\mathrm{D}$ is slightly slower than the benchmarks, all methods satisfy the real-time speed requirement for online sensing.

The average values of the F-measure and the ER against the iteration number (number of sampled points) are also plotted in Figure 5.4. From this figure, we can conclude that the F-measure of $\mathrm{AKM}^2\mathrm{D}$ is strictly better than Grid and Random methods for any number of sampled points. Furthermore, the ER of $\mathrm{AKM}^2\mathrm{D}$ increases to 18% with only 200 points

Table 5.1: Anomaly Detection Result with 250 and 400 sampled points

Methods	250 sampled points			400 sampled points		
	AKM^{2}D	Random	Grid	AKM^{2}D	Random	Grid
Precision	0.6895	0.8081	0.8012	0.7498	0.7843	0.6681
Recall	0.7802	0.2662	0.2726	0.8816	0.4020	0.6492
F	0.7212	0.3815	0.3930	0.8047	0.5088	0.6539
ER	18.31%	5.37%	5.62%	17.81%	5.36%	14.17%
AMMD	0.0367	0.0735	0.0545	0.0272	0.0631	0.0495
MMD	0.0681	0.1187	0.0700	0.0560	0.0952	0.0699
Time	0.0046s	0.0026s	0.0025s	0.0053s	0.0028s	0.0032s

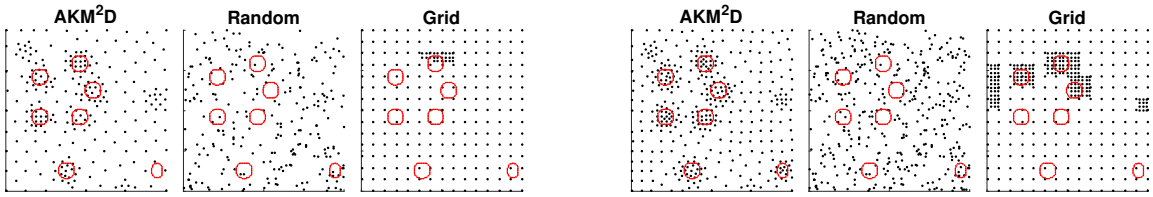


(a) F-measure (b) Exploration ratio

Figure 5.4: F-measure and Exploitation Ratio

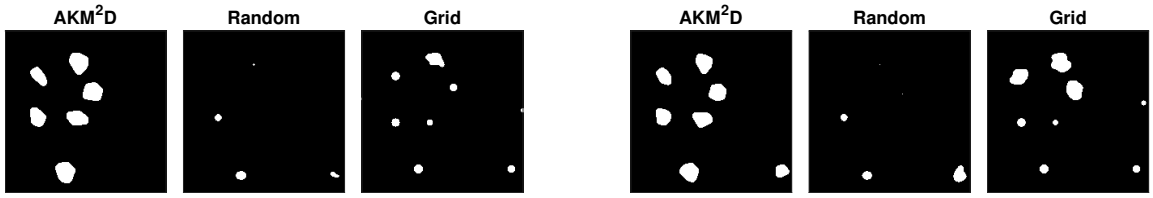
and then oscillating around 18%, showing its superiority to quickly locate and sample the anomalous regions. The ER of Grid stays at 4% during the coarse grid sampling and only begin to increase up to 16% when performing the fine-grid sampling (after 225 points). Finally, the ER of Random stays the same as 5.8%, which is the percentage of true anomalous regions.

Furthermore, we investigate the pattern of sampled points (with 250 and 400 points) in Figure 5.5. From the figure, we can observe that with only 250 sampled points, AKM^{2}D discovers all anomalous regions but one, with a better space-filling point distribution. However, Random fails to detect any of the anomalous regions and Grid can only detect one. On the other hand, 400 sampled points are enough for AKM^{2}D to detect all 7 anomalous regions. However, again Random fails to discover any anomalous regions and Grid finishes



(a) 250 sampled points (b) 400 sampled points

Figure 5.5: Sampled point pattern for all methods for 250 and 400 points



(a) 250 sampled points (b) 400 sampled points

Figure 5.6: Anomaly estimation result for all methods for 250 and 400 points

with the fine-grid sampling of only three regions. Also, we plot the detected anomalies corresponding to 250 and 400 sampled points in Figure 5.6, which again indicates the superior performance of $\mathrm{AKM}^2\mathrm{D}$ in anomaly detection.

5.5 Case Study

In this section, the proposed adaptive sampling and estimation framework is applied to a real dataset in the NDE area. The case study pertains to anomaly detection in composite laminates using a guided wave-field (GW) inspection system. Lamb wave-based inspection is one of the popular methods in NDE and structural health monitoring due to its high sensitivity to detecting anomalies invisible to the naked eye [14]. However, existing GW techniques are point-based and require the whole-field inspection of a specimen. The whole-field inspection is typically a time-consuming process as it requires sensing of a large number of points to avoid spatial aliasing and to achieve the desired resolution [14]. Therefore, it is vital to reduce the data acquisition time by reducing the number of sampled points using an adaptive sampling strategy.

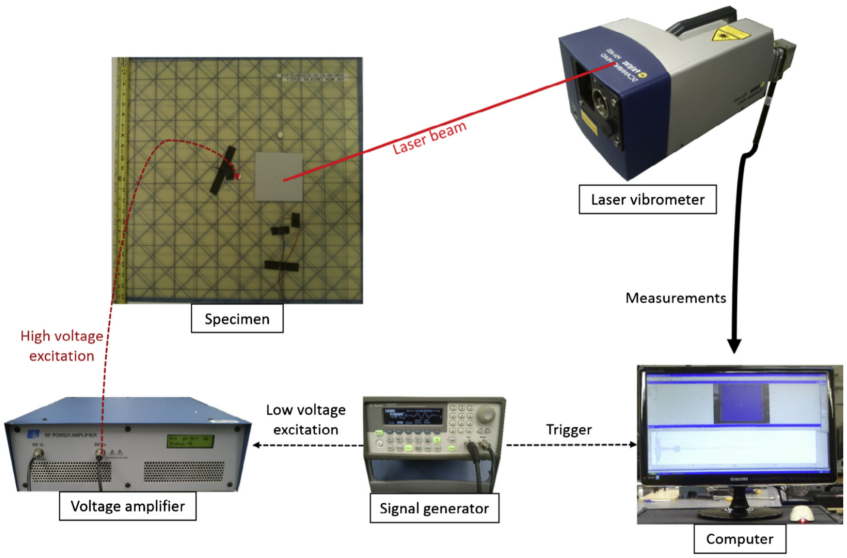
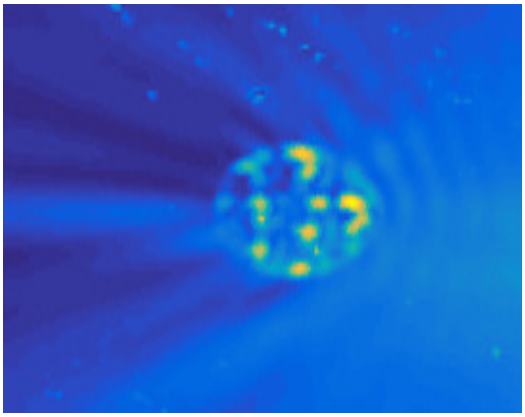


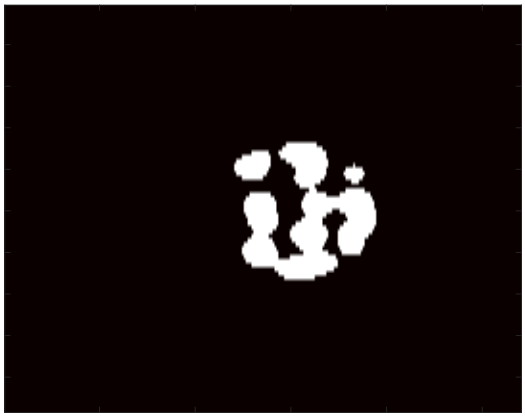
Figure 5.7: Guided wavefield experiment setup [14]

The setup of our GW experiment is shown in Figure 5.7. A scanning laser Doppler vibrometer (SLDV) is employed for wavefield measurement over a grid of points with the resolution of 270×100. It takes around 2 hours to inspect a $600 \times 600 \times 1.6$ mm composite laminate with 8 layers. The specimen contains several artificial delaminations in the center as shown in Figure 5.8a, which is the energy map of the entire wavefield based on complete sampling. To speed up the GW test so that it can be used for online inspection, we reduce the number of sensing points by using adaptive sampling strategies. For comparison purposes, we show detected anomalies using complete sampling strategy (i.e. Figure 5.8a) in Figure 5.8b. The objective is to achieve a similar detection accuracy with the least number of sampled points.

We apply $\mathrm{AKM}^2\mathrm{D}$ as well as two other benchmark methods (i.e. Random and Grid) for adaptive sampling and use the proposed estimation methods for anomaly detection. We compare the detection results obtained from the adaptive sampling methods with those of the complete sampling, shown in Figure 5.8b, (as the ground truth), and compute the F-measure and ER profiles depicted in Figure 5.9. We can observe that with only 300 points (1.1% of complete sensing) $\mathrm{AKM}^2\mathrm{D}$ is able to achieve the F-measure of 0.8 much higher

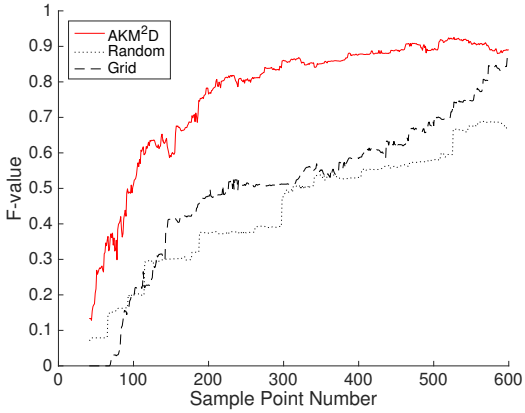


(a) Energy map of the entire wavefield

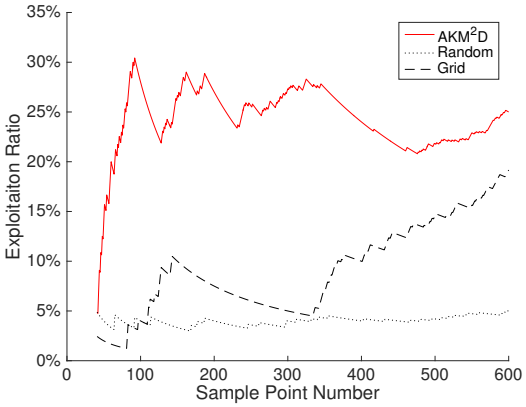


(b) Detected anomaly

Figure 5.8: Energy map of the entire wavefield and detected anomaly



(a) F-measure

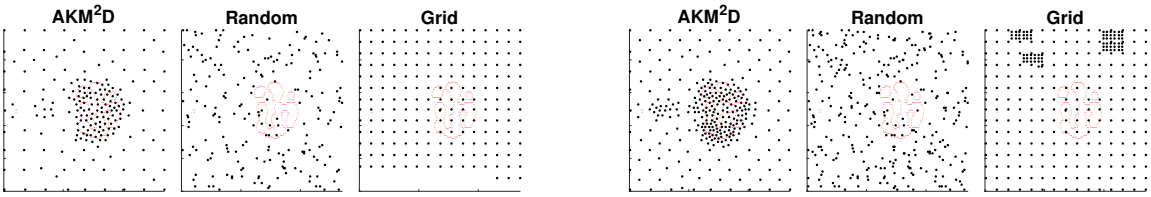


(b) Exploitation Ration

Figure 5.9: F-measure and Exploitation Ration

than those of Random and Grid around 0.5.

The pattern of sampled points and detected anomalous regions by using 200 and 300 points are also shown in Figures 5.10 and 5.11, respectively. From these figures, it is clear that, the irregular anomalous regions can be fully explored by the proposed $\mathrm{AKM}^2\mathrm{D}$ with only 200 sampled points (0.7% of full sampling), which can reduce the measurement time from 2 hours to a few seconds. However, using Random and Grid methods, very few sampled points are selected in the anomalous regions.



(a) 200 sampled points (b) 300 sampled points

Figure 5.10: Sampled point pattern for all methods for 200 and 300 points

(a) 200 sampled points (b) 300 sampled points

Figure 5.11: Anomaly estimation result for all methods for 200 and 300 points

5.6 Conclusion

Adaptive sampling for clustered anomaly detection is vital in scaling up point-based inspection systems. In this chapter, we proposed a novel methodology for real-time adaptive sampling and anomaly detection in large sampling spaces. In our methodology, we first developed an adaptive sampling framework, namely the $\mathrm{AKM}^2\mathrm{D}$, by optimizing a composite index. We also studied the sampling properties and showed that the proposed method is able to balance sampling between the exploration of the entire space and the focused sampling near anomalies. We developed efficient and recursive algorithms to determine the location of the next sampled point by solving the optimization problem in real time. Then, we proposed robust kernel regression and sparse kernel regression to update the estimates of the functional mean and the anomalous regions after a new sample is collected. In the simulation study, we showed that the proposed $\mathrm{AKM}^2\mathrm{D}$ outperforms existing adaptive sampling approaches, which fail to locate and focus on anomalous regions. Finally, the proposed method was applied to a real case study on the anomaly detection of composite laminates via guided wavefield test. We showed that our method can achieve a similar

detection accuracy to that of the complete sampling by sensing only 0.7% of the sampled points, and hence it can significantly reduce the inspection time. There are several potential research directions to be investigated.

CHAPTER 6

POINT CLOUD DATA MODELING AND ANALYSIS VIA REGULARIZED TENSOR REGRESSION AND DECOMPOSITION

In this chapter, we represent point clouds using tensors and propose regularized tucker decomposition and regularized tensor regression to model the variational patterns of point clouds and link them to process variables. The performance of the proposed method is evaluated through simulation and a real case study of turning process optimization.

The remainder of the chapter is organized as follows. Section 6.1 briefly reviews the basic tensor notation and multilinear algebra. Section 6.2 first introduces the general regression framework for tensor response data and then elaborates the two frameworks for basis selection, i.e., RTDR and RTR. Section 6.3 validate the proposed methodology by using simulated data with two different types of structured point clouds. In this section, the performance of the proposed methods is compared with some existing two-step methods in terms of estimation accuracy. In Section 6.4, we illustrate a case study for process modeling and optimization in a turning process. Finally, we conclude the paper with a short discussion in Section 6.5.

6.1 Basic Tensor Notation and Multilinear Algebra

In this section, we introduce basic notations, definitions, and operators in multilinear (tensor) algebra that we use in this paper. Throughout the paper, scalars are denoted by lowercase italic letters, e.g., a, vectors by lowercase boldface letters, e.g., $\mathbf{a}$, matrices by uppercase boldface letter, e.g., $\mathbf{A}$, and tensors by calligraphic letters, e.g., $\mathcal{A}$. For example, an order-K tensor is represented by $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_K}$, where I_k represent the mode-k dimension of $\mathcal{A}$. The mode-k product of a tensor $\mathcal{A}$ by a matrix $\mathbf{V} \in \mathbb{R}^{P_k \times I_k}$ is defined by $(\mathcal{A} \times_k \mathbf{V})(i_1, \cdots, i_{n-1}, j_k, i_{n+1}, \cdots, i_K) = \sum_{i_k} A(i_1, \cdots, i_k, \cdots, i_N) V(j_k, i_k)$. The Frobenius norm of a tensor $\mathcal{A}$ can be defined as $\|\mathcal{A}\|_F^2 = \sum_{i_1, \cdots, i_K} A(i_1, \cdots, i_k, \cdots, i_K)^2$.

The n-mode unfold maps the tensor $\mathcal{A}$ into matrix $\mathbf{A}_{(n)}$, where the column of $\mathbf{A}_{(n)}$ are the n-mode vectors of $\mathcal{A}$.

Tucker decomposition decomposes a tensor into a core tensor multiplied by a matrix along each mode, $\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_K \mathbf{U}^{(K)}$, where $\mathbf{U}^{(k)}$ is an orthogonal $I_k \times I_k$ matrix and is the principal components in each mode. Tensor product can be represented equivalently by Kronecker product $\mathrm{vec}(\mathcal{A}) = (\mathbf{U}^{(K)} \otimes \cdots \otimes \mathbf{U}^{(1)})\mathrm{vec}(\mathcal{S})$, where vec is the vectorized operator defined as $\mathrm{vec}(\mathcal{A}) = \mathbf{A}_{(K+1)}$ (a $I_1 \cdots I_K$-dimension vector). The definition of Kronecker product is as follow: Suppose $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$ are matrices, the Kronecker product of these matrices, denoted by $\mathbf{A} \otimes \mathbf{B}$, is an $mq \times nq$ block matrix defined by $\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$.

6.2 Tensor Regression Model with Scalar Input

In this paper, to simplify presentation, we demonstrate the methodology with 2D response variable. However, this method can be easily extended to higher order tensors by simply adding other dimensions. Suppose a training sample of size N is available that includes tensor responses denoted by $\mathbf{Y}_i \in \mathbb{R}^{I_1 \times I_2}$, $i = 1, \cdots, N$ along with the corresponding input variables denoted by $\mathbf{x}_i \in \mathbb{R}^{p \times 1}$, $i = 1, \cdots, N$, where p is the number of regression coefficients. The tensor regression aims to link the response $\mathbf{Y}_i$ with the input variables $\mathbf{x}_i$ through a tensor coefficient $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times p}$ such that

$$\mathbf{Y}_i = \mathcal{A} \times_3 \mathbf{x}_i + \mathbf{E}_i, i = 1, \cdots, N \tag{6.1}$$

where $\mathbf{E}_i \overset{iid}{\sim} N(0, \sigma^2)$ represents the random noises. We can combine the response data $\mathbf{Y}_i$ and the residual $\mathbf{E}_i$ in 3D tensors as $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times N}$ and $\mathcal{E} \in \mathbb{R}^{I_1 \times I_2 \times N}$, respectively. Furthermore, we combine all $\mathbf{x}_i$ in a single input matrix $\mathbf{X} \in \mathbb{R}^{\mathrm{N} \times \mathrm{p}}$. Therefore, (6.1) can

be represented compactly in the tensor format as shown in (6.2).

$$\mathcal{Y} = \mathcal{A} \times_3 \mathbf{X} + \mathcal{E} \tag{6.2}$$

Intuitively, the coefficient tensor in (6.2) can be estimated by using the least square estimation method, i.e.,

$$\hat{\mathcal{A}} = \underset{\mathcal{A}}{\operatorname{argmin}} \|\mathcal{Y} - \mathcal{A} \times_3 \mathbf{X}\|_F^2, \tag{6.3}$$

which has a closed form solution in the form of $\mathbf{A}_{(3)} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y}_{(3)})$, where $\mathbf{Y}_{(3)}$ and $\mathbf{A}_{(3)}$ are the mode-3 unfolding of $\mathcal{Y}$ and $\mathcal{A}$, respectively. However, since the dimension of $\mathcal{A}$ is too high, solving (6.3) directly could result in severe overfitting. A common procedure is to assume that the coefficient tensor $\mathcal{A}$ is low rank and hence it can be represented in a low-dimensional functional space expanded by basis $\mathbf{U}^{(k)}, k = 1, 2$, as shown in (6.4).

$$\mathcal{A} = \mathcal{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} + \mathcal{E}_{\mathcal{A}} \tag{6.4}$$

where $\mathcal{E}_{\mathcal{A}}$ is the residual tensor of projecting the coefficient $\mathcal{A}$ into the low dimensional space. $\mathcal{B} \in \mathbb{R}^{P_1 \times P_2 \times p}$ is the core tensor in the low dimensional space after the projection. If $\mathbf{U}^{(k)}$ is complete, the residual tensor $\|\mathcal{E}_{\mathcal{A}}\|_F = 0$. In most applications, however, $\mathcal{A}$ lies in a low-dimensional space, which is known as the "Blessing of Dimensionality". Therefore, by using a low-dimensional basis $\mathbf{U}^{(k)}$ (i.e., $P_k \ll I_k$), we can significantly reduce the dimensionality of the coefficient tensor $\mathcal{A}$ and still have $\|\mathcal{E}_{\mathcal{A}}\|_F$ close to zero. Since $\mathcal{E}_{\mathcal{A}}$ is negligible, when $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are given, $\hat{\mathcal{A}}$ can be approximated by $\hat{\mathcal{B}} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}$, where $\hat{\mathcal{B}}$ can be estimated by solving the following tensor regression formulation (6.5).

$$\hat{\mathcal{B}} = \underset{\mathcal{B}}{\operatorname{argmin}} \|\mathcal{Y} - \mathcal{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{X}\|_F^2 \tag{6.5}$$

Proposition 11. *the optimization problem* (6.5) *has a closed-form solution that can be*

expressed by

$$\hat{\mathcal{B}} = \mathcal{Y} \times_1 (\mathbf{U}^{(1)^T}\mathbf{U}^{(1)})^{-1}\mathbf{U}^{(1)^T} \times_2 (\mathbf{U}^{(2)^T}\mathbf{U}^{(2)})^{-1}\mathbf{U}^{(2)^T} \times_3 (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{6.6}$$

The proof of proposition 11 is shown in Appendix D.

The choice of basis $\mathbf{U}^{(k)}, k = 1, 2$ is important and for a more accurate model, they should be carefully chosen. In the next subsections, we propose two methods for defining basis; one method is based on a data-driven approach and the other one incorporates the user knowledge about the process and response. Specifically, in the first method, $\mathbf{U}^{(k)}, k = 1, 2$ are extracted from data by applying regularized Tucker decomposition and second method shows how a user-defined basis such as B-spline can be used.

6.2.1 Regularized Tucker Decomposition

In this section, we propose a new regularized Tucker decomposition, which is capable of extracting smooth variational patterns from a tensor response. We then demonstrate how it can be integrated with the tensor regression model presented in (6.5).

Tucker Decomposition Regression

Principal component analysis (PCA) [81] has been widely used because of its ability to reduce the dimensionality of high-dimensional data. However, as pointed out by [3], applying PCA directly on tensor data requires to unfold the original tensor into a long vector, which may result in the loss of the structural information of the original tensor. To overcome this difficulty, tensor decomposition techniques such as Tucker Decomposition [118] have been proposed and widely applied in image denoising, image monitoring, tensor completion, etc. Tucker decomposition aims to find a set of orthogonal transformation matrices $\mathbf{U} = \{\mathbf{U}^{(k)} \in \mathbb{R}^{I_k \times P_k}; \mathbf{U}^{(k)^T}\mathbf{U}^{(k)} = \mathbf{I}_{P_k}, P_k < I_k, k = 1, 2\}$ such that it can best represent

the original data $\mathcal{Y}$, where $\mathbf{I}_{P_k}$ represents the identity matrix of size $P_k \times P_k$. That is,

$$\{\hat{\mathcal{S}}, \hat{\mathbf{U}}^{(1)}, \hat{\mathbf{U}}^{(2)}\} = \underset{\mathcal{S},\mathbf{U}^{(1)},\mathbf{U}^{(2)}}{\operatorname{argmin}} \|\mathcal{Y} - \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}\|_F^2 \tag{6.7}$$

$\hat{\mathcal{S}}$ is the core tensor and can be obtained by

$$\hat{\mathcal{S}} = \mathcal{Y} \times_1 \hat{\mathbf{U}}^{(1)^T} \times_2 \hat{\mathbf{U}}^{(2)^T} \tag{6.8}$$

[3] showed that (6.7) is equivalent to maximize the variation of the projected low-dimensional tensor, known as Multi-linear Principal Component Analysis (MPCA) method proposed in [119]. Therefore, for finding the basis matrix, one can solve the following optimization problem.

$$\{\hat{\mathbf{U}}^{(1)}, \hat{\mathbf{U}}^{(2)}\} = \underset{\mathbf{U}^{(1)},\mathbf{U}^{(2)}}{\operatorname{argmax}} \|\mathcal{Y} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T}\|_F^2 \tag{6.9}$$

Moreover, it follows directly from Proposition 11, (6.8) and (6.6) that (6.5) can be solved efficiently by regressing the core tensor $\hat{\mathcal{S}}$ on the input variables matrix $\mathbf{X}$.

$$\hat{\mathcal{B}} = \hat{\mathcal{S}} \times_1 (\hat{\mathbf{U}}^{(1)^T}\hat{\mathbf{U}}^{(1)})^{-1} \times_2 (\hat{\mathbf{U}}^{(2)^T}\hat{\mathbf{U}}^{(2)})^{-1} \times_3 (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{6.10}$$

Furthermore, if $\hat{\mathbf{U}}^{(k)}$ is orthogonal, it is easy to show that

$$\hat{\mathcal{B}} = \hat{\mathcal{S}} \times_3 (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{6.11}$$

Therefore, for the orthogonal basis, directly solving (6.5) for $\hat{\mathcal{B}}$ is equivalent to the following two-step approach: 1) Apply Tucker decomposition on original tensor $\mathcal{Y}$ to compute the basis $\mathbf{U}^{(k)}$ and core tensor $\hat{\mathcal{S}}$. 2) Regress the core tensor $\hat{\mathcal{S}}$ on the input variable $\mathbf{X}$, where the core tensor $\hat{\mathcal{S}}$ is computed by (6.8).

Regularized Tucker Decomposition

As shown in [120], in high-dimensional cases, if the structural information of the eigenbasis such as smoothness or sparsity is incorporated in decomposition procedure, more accurate estimates of the eigenbasis can be obtained. Inspired by smoothed functional principal components analysis[121], in this section, we propose a new regularized Tucker decomposition to penalize the roughness of the eigenbasis by changing the orthogonality of the basis, as shown in the following equation:

$$\{\hat{\mathbf{U}}^{(1)}, \hat{\mathbf{U}}^{(2)}\} = \underset{\mathbf{U}^{(1)},\mathbf{U}^{(2)}}{\operatorname{argmax}} \|\mathcal{Y} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T}\|_F^2 \tag{6.12}$$
$$s.t. \quad \mathbf{U}^{(k)^T}\mathbf{R}_k(\lambda)\mathbf{U}^{(k)} = \mathbf{I}_{P_k}, k = 1, 2$$

where $\mathbf{R}_k(\lambda) = \mathbf{I}_{I_k} + \lambda(\mathbf{D}_k^2)^T\mathbf{D}_k^2$ is the roughness matrix to control the level of smoothness of eigenbasis $\mathbf{U}^{(k)}$, where $\mathbf{D}_k$ is the first order difference operator. For open boundary conditions, $\mathbf{D}_k = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix}$, For periodic boundary condition, $\mathbf{D}_k = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \\ -1 & & & 1 \end{bmatrix}$. It should be noted that if $\lambda = 0$, the regularized Tucker decomposition (6.12) becomes the traditional Tucker decomposition in (6.7). Similarly to the traditional Tucker Decomposition, we can prove in Proposition 12 that (6.12) is equivalent to minimize the reconstruction error with the smoothness penalty.

Proposition 12. *Regularized Tucker Decomposition* (6.12) *is equivalent to the following penalized tensor regression* (6.13) *in the sense that*

$$\{\hat{\mathbf{U}}^{(1)}, \hat{\mathbf{U}}^{(2)}, \hat{\mathcal{S}}\} = \underset{\mathbf{U}^{(1)},\mathbf{U}^{(2)},\mathcal{S}}{\operatorname{argmax}} \|\mathcal{Y} - \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}\|_F^2 + s^T\mathbf{P}_s s \tag{6.13}$$

Algorithm 9: ALS algorithm

initialize

$\mathbf{R}_k(\lambda) = I + \lambda(D_k^2)^T D_k^2, k = 1, 2$

Compute Chelosky Decomposition: $\mathbf{L}_k^T \mathbf{L}_k = \mathbf{R}_k(\lambda)$, $\mathbf{S}_k = \mathbf{L}_k^{-1}$, $\mathbf{V}_k = \mathbf{S}_k^T \mathbf{W}_k \mathbf{S}_k, k = 1, 2$

end

for $i = 1, \cdots n_{iter}$ **do**

Solve the eigenvalue problem: $\mathbf{V}_k \mathbf{Z}^{(k)} = \mathbf{Z}^{(k)} \mathbf{\Lambda}_k, k = 1, 2$

$\mathbf{U}^{(k)} = \mathbf{S}_k \mathbf{Z}^{(k)}$

end

and $\hat{\mathcal{S}} = \mathcal{Y} \times_1 \hat{\mathbf{U}}^{(1)^T} \times_2 \hat{\mathbf{U}}^{(2)^T}$ *where* $s = \text{vec}(\mathcal{S})$ *and* $\mathbf{P}_s = (\lambda \mathbf{U}_2^T \mathbf{U}_2 \otimes \mathbf{U}_1^T (\mathbf{D}_1^2)^T \mathbf{D}_1^2 \mathbf{U}_1 + \lambda \mathbf{U}_2^T (\mathbf{D}_2^2)^T \mathbf{D}_2^2 \mathbf{U}_2 \otimes I_1 \mathbf{U}_2 + \lambda^2 \mathbf{U}_2^T (\mathbf{D}_2^2)^T \mathbf{D}_2^2 \mathbf{U}_2 \otimes \mathbf{U}_1^T (\mathbf{D}_1^2)^T \mathbf{D}_1^2 \mathbf{U}_1)$ *is the roughness matrix to encourage the smoothness of* $\mathbf{U}_k$.

Proof is given in the appendix B.

Similar to the traditional Tucker decomposition, Alternative Least Square (ALS) algorithm can be used to update $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ iteratively until convergence. We then showed in Proposition 13 that each update of $\mathbf{U}^{(k)}, k = 1, 2$ yields a close-form solution.

Proposition 13. *Given* $\mathbf{U}^{(-k)}$*, the solution of* $\text{argmax}_{\mathbf{U}^{(k)}} \|\mathcal{Y} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T}\|_F^2$ *is given by the following generalized eigenvalue problem*

$$\mathbf{W}_k \mathbf{U}^{(k)} = \mathbf{R}_k(\lambda) \mathbf{U}^{(k)} \Lambda_k$$

where $\mathbf{W}_k = \mathbf{Y}_{(k)} \mathbf{U}^{(-k)} \mathbf{U}^{(-k)^T} \mathbf{Y}_{(k)}^T$, Λ_k *is the diagonal eigenvalue matrix, and* $\mathbf{U}^{(-k)} = \begin{cases} \mathbf{U}^{(2)} & k = 1 \\ \mathbf{U}^{(1)} & k = 2 \end{cases}$.

The proof is given in Appendix D. The fact that the subproblem in each iteration reduces to the generalized eigenvalue problem significantly speeds up the ALS algorithm. The detailed of the ALS algorithm is given in Algorithm 9.

6.2.2 Customized Basis Selection

In other cases, we would like to customize the basis in (6.5) to represent the tensor response based on domain knowledge or other data characteristics. For example, a pre-defined spline or kernel basis can be used to represent general smooth tensors. Fourier basis or periodic B-spline basis can be used to represent smooth tensors with periodic boundary constrain. Furthermore, penalization can be added to (6.5) to control the level of smoothness as

$$\hat{\mathcal{B}} = \underset{\mathcal{B}}{\operatorname{argmin}} \|\mathcal{Y} - \mathcal{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{X}\|_F^2 + P(\mathcal{B}) \tag{6.14}$$

(6.5) can be represented alternatively by the Kronecker product as $\hat{\beta} = \operatorname{argmin}_\beta \|\mathbf{y} - (\mathbf{X} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})\boldsymbol{\beta}\|^2 + P(\boldsymbol{\beta})$, which is in the form of a regression problem. However, the dimensions of $\mathbf{X} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)}$ in (6.15) is $\mathbb{R}^{NI_1I_2 \times pP_1P_2}$, which is often too large to compute or even stored for high dimensional problems. Therefore, to address this computational challenge, following [72], we use a special from of the penalty term defined by

$$P(\boldsymbol{\beta}) = \boldsymbol{\beta}^T(\mathbf{X}^T\mathbf{X}) \otimes (\lambda\mathbf{P}_2 \otimes \mathbf{U}^{(1)^T}\mathbf{U}^{(1)} + \lambda\mathbf{U}^{(2)^T}\mathbf{U}^{(2)} \otimes \mathbf{P}_1 + \lambda^2\mathbf{P}_2 \otimes \mathbf{P}_1)\boldsymbol{\beta} \tag{6.15}$$

where $\boldsymbol{\beta} = \operatorname{vec}(\mathcal{B})$, $\mathbf{P}_k = (\mathbf{D}_k^2)^T\mathbf{D}_k^2$ is the penalization matrix to control the smoothness of mode-k of the original tensor. It has shown in [96] and [72] that the penalty term defined with tensor structure works well in simulation and achieve optimal rate of convergence asymptotically under some mild conditions.

We proved in Proposition 14 that by using this $P(\mathcal{B})$, Problem (6.14) becomes separable for different modes of the original tensor.

Proposition 14. *If $P(\mathcal{B})$, defined in* (6.15)*, is used in the optimization problem* (6.14)*, can*

be solved efficiently via tensor product by

$$\hat{\mathcal{B}} = \mathcal{Y} \times_1 (\mathbf{U}^{(1)^T}\mathbf{U}^{(1)} + \lambda\mathbf{P}_1)^{-1}\mathbf{U}^{(1)^T} \times_2 (\mathbf{U}^{(2)^T}\mathbf{U}^{(2)} + \lambda\mathbf{P}_2)^{-1}\mathbf{U}^{(2)^T} \times_3 (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{6.16}$$

The proof is given in Appendix D. Finally, to select the tuning parameter λ in the algorithm, the GCV criterion can be used, where the tuning parameter λ can be selected by solving $\hat{\lambda} = \arg\min_\lambda \mathrm{GCV}(\lambda) = \arg\min_\lambda \frac{\|\mathcal{Y}-\hat{\mathcal{B}}\times_1\mathbf{U}^{(1)}\times_2\mathbf{U}^{(2)}\times_3\mathbf{X}\|^2/n}{(1-n^{-1}\mathrm{tr}(\hat{\mathbf{H}}_1(\lambda))\mathrm{tr}(\hat{\mathbf{H}}_2(\lambda))\mathrm{tr}(\hat{\mathbf{H}}_3(\lambda)))^2}$, where $\hat{\mathbf{H}}_k(\lambda) = \mathbf{U}^{(k)}(\mathbf{U}^{(k)^T}\mathbf{U}^{(k)} + \lambda\mathbf{P}_k)^{-1}\mathbf{U}^{(k)^T}$, $k = 1, 2$, and $\hat{\mathbf{H}}_3(\lambda) = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$.

6.3 Simulation Study

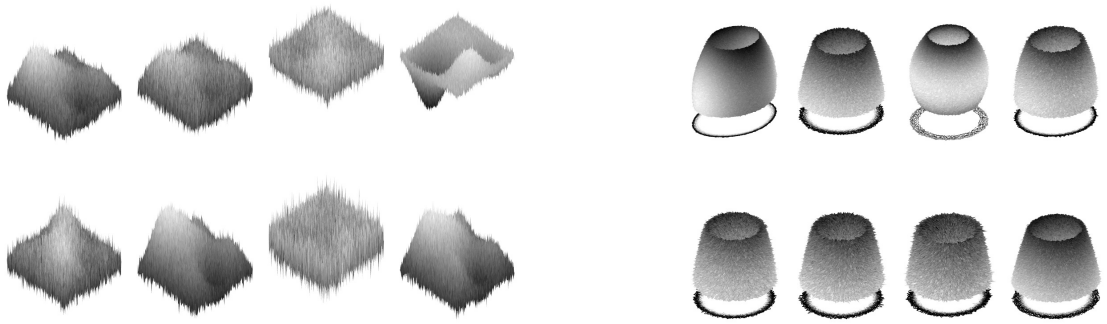
In this section, we conduct simulations to evaluate the proposed Regularized Tucker Decomposition Regression (RTDR) and Regularized Tensor Regression (RTR) for structured point cloud modeling. We simulate N structured point cloud as training samples $\mathbf{Y}_i, i = 1, \cdots, N$ with two different scenarios (i.e. surface shape and truncated cone shape) by following $\mathbf{Y}_i = \mathbf{M} + \mathbf{V}_i + \mathbf{E}_i$, or equivalently in the tensor format $\mathcal{Y} = \mathcal{M} + \mathcal{V} + \mathcal{E}$, where $\mathcal{Y}$ is the 3rd order tensor combining $\mathbf{Y}_i, i = 1, \cdots, N$; $\mathcal{M}$ is the mean of the point cloud data and $\mathcal{V}$ is the variational pattern of the point cloud due to different input variables $\mathbf{x}_i$. Finally, random noise $\mathbf{E}_i$ is generated by $\mathbf{E}_i \overset{i.i.d}{\sim} N(0, \sigma^2)$.

Case 1. Surface point cloud simulation In this case, we simulate the surface point cloud in a 3D Cartesian coordinate system (x, y, z) where $0 \leq x, y \leq 1$. The corresponding $z_{i_1 i_2}$ value at $(\frac{i_1}{I_1}, \frac{i_2}{I_2}), i_1 = 1, \cdots, I_1; i_2 = 1, \cdots I_2$, with $I_1 = I_2 = 200$ for i^{th} sample is recorded in a matrix $\mathbf{Y}_i$. We then simulate 100 training samples with variational patterns of point cloud surface $\mathcal{V}$ according to the following linear model $\mathcal{V} = \mathcal{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{X}$. In the simulation setup, we select three basis $\mathbf{U}^{(k)} = [\mathbf{u}_1^{(k)}, \mathbf{u}_2^{(k)}, \mathbf{u}_3^{(k)}]$ with $\mathbf{u}_\alpha^{(k)} = [\sin(\frac{\pi\alpha}{n}), \sin(\frac{2\pi\alpha}{n}), \cdots, \sin(\frac{n\pi\alpha}{n})]^T, \alpha = 1, 2, 3$. The two mode-3 slices of $\mathcal{B} \in \mathbb{R}^{3\times3\times2}$ is

generated as $\mathbf{B}_1 = \begin{bmatrix} 4 & 1 & 0 \\ 1 & 0.1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$, $\mathbf{B}_2 = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 3 & 0 \\ 1 & 0 & 0.2 \end{bmatrix}$. The examples of the generated point cloud surface are shown in Figure 6.1a.

Case 2. Truncated cone point cloud simulation In this case, we simulate truncated cone point clouds in the 3D cylindrical coordinate system (r, ϕ, z), where $\phi \in [0, 2\pi]$, $z \in [0, 1]$. The corresponding r value at $(\phi, z) = (\frac{2\pi i_1}{I_1}, \frac{i_2}{I_2})$, $i_1 = 1, \cdots, I_1; i_2 = 1, \cdots I_2$ with $I_1 = I_2 = 200$ for i^{th} sample is recorded in the matrix $\mathbf{Y}_i$. We simulate the variational patterns of point cloud surface $\mathcal{V}$ according to $r(\phi, z) = \frac{r_0 + z\tan\theta}{\sqrt{1-e^2\cos^2\phi}} + c(z^2 - z)$ with different settings of θ, r_0, e, c as follow: 1) different angles of the cone, i.e., $\theta \in \{0, \frac{\pi}{8}, \frac{\pi}{4}\}$; 2) different radii of the upper circle, i.e., $r_0 \in \{1.1, 1.3, 1.5\}$; 3) different eccentricities of top and bottom surfaces, i.e., $e \in \{0, 0.3, 0.5\}$; 4) different curvatures of the side of the truncated cone, i.e., $c = \{-1, 0, 1\}$. Finally, we conduct a full factorial design to generate $3^4 = 81$ training samples with different combinations of these coefficients. Furthermore, we define four input variables by $x_1 = \tan\theta$, $x_2 = r$, $x_3 = e^2$, $x_4 = c$ and record them in an input matrix $\mathbf{X}$ of size 81×4. These nonlinear transformations lead to a better linear approximation of the point cloud in the cylindrical coordinate system with the input matrix $\mathbf{X}$. The examples of the generated truncated cone are shown in Figure 6.1b. Finally, we generated 1000 testing examples $\mathcal{Y}_{te}$ based on the variational patterns generated from $\theta \sim U(0, \frac{\pi}{4})$, $r \sim U(1.1, 1.5)$, $e \sim U(0, 0.5)$, $c \sim U(-1, 1)$, where U denotes the uniform distribution.

For both cases, The goal is to find the relationship between the point cloud tensor $\mathcal{Y}$ and input variables $\mathbf{X}$. We compare our proposed Regularized Tensor Regression (RTR) and Regularized Tucker Decomposition Regression (RTDR) with two existing methods in the literature. The benchmark methods we used for comparison include Vectorized Principal Component Analysis (VPCA) and simple linear regression (LR). For both benchmark methods, smoothing is first applied as a preprocessing step to remove the noise. For VPCA,



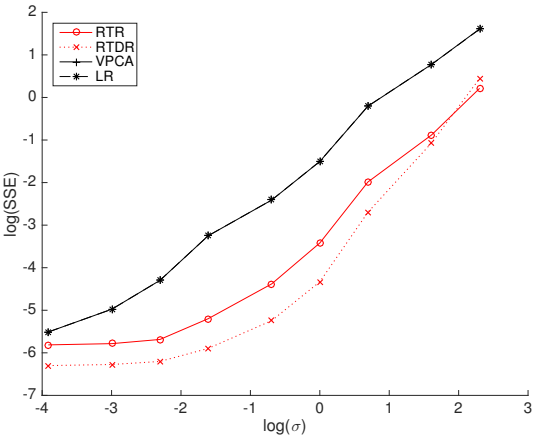
(a) Case 1: Examples of generated surface

(b) Case 2: Examples of generated truncated cone

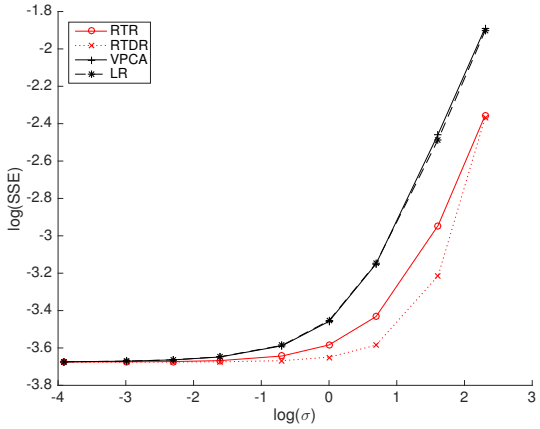
Figure 6.1: Examples of generated point cloud for simulation study

PCA is applied on the unfolded matrix denoted by $\mathbf{Y}_{(3)}$. For LR, we conduct a linear regression for each entry of the tensor $\mathcal{Y}$ with the input variables $\mathbf{X}$, separately. For RTR, we use B-spline with 10 knots on each dimension. It should be noted that in Case 2, we apply the periodic B-spline with period 2π to model the periodicity in the θ direction. Similarly for RTDR, we apply the periodic difference matrix D_k in Case 2. The tuning parameters of the RTR and RTDR are selected by using the GCV criterion and cross validation. Finally, for Case 1, the sum of squared error (SSE) between the true coefficients $\mathcal{A} = \mathcal{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}$ and estimated coefficients $\hat{\mathcal{A}}$ is evaluated. For Case 2, since we generate complex variational patterns, the sum of squared error (SSE) between $\mathcal{Y}_{te}$ and the predicted tensor $\hat{\mathcal{Y}}_{te}$ is evaluated from 10000 simulation replications under different noise levels σ, shown in Figure 6.2.

From Figure 6.2, we can conclude that the SSEs of the VPCA and LR (after smoothing) is much larger than the SSEs of the proposed RTR and RTDR. The main reason is that VPCA and LR do not consider the tensor structure and spatial structure of the simulated point clouds. RTDR works better especially when the noise level is low. The reason is that the smoothness of RTDR is controlled by the regularization term, which is more flexible than RTR, where the functional space is constructed by the predefined B-spline basis. Finally, the estimated coefficient $\hat{\mathcal{A}}$ of both the proposed methods and true coefficients for

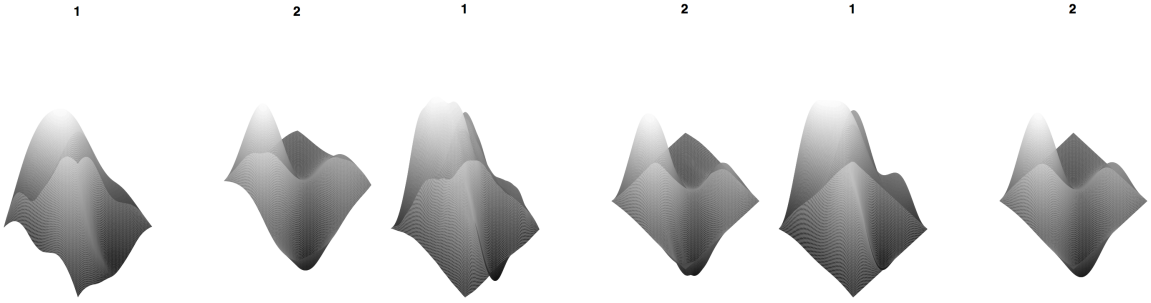


(a) Case 1: SSE of $\mathcal{A}$ with magnitude of noise σ



(b) Case 2: SSE of $\mathcal{Y}_{te}$ with magnitude of noise σ

Figure 6.2: SSE of the proposed methods with different magnitude in Case 1 and Case 2



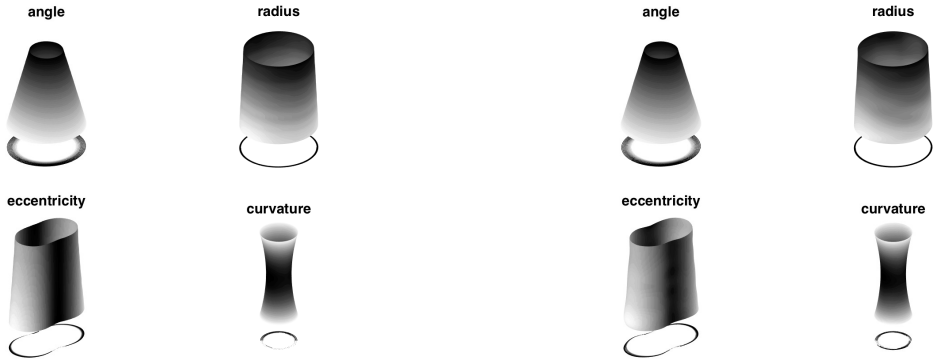
(a) Tensor Regression (b) Tucker Decomposition (c) True coefficient

Figure 6.3: Estimated and true coefficient for case 1

Case 1 are plotted in Figure 6.3. We can conclude that both RTR and RTDR are capable of estimating the true coefficient accurately. Furthermore, in Case 2, if the noise magnitude is very small, the SSEs of all methods reduce to the linear approximation error of the complex point cloud. The estimated coefficient $\hat{\mathcal{A}}$ for both cases of the proposed RTR and RTDR are shown in Figure 6.4. From this figure, we can conclude that both methods are able to estimate the surface coefficients accurately.

6.4 Case Study

In this section, the proposed RTR and RTDR methods are applied to a case study in the turing process, in which cylinders of titanium alloy Ti-6A1-4V were machined from an



(a) Tensor regression (b) Tucker decomposition

Figure 6.4: Estimated coefficient for case 2

initial 20 mm diameter to the diameter of 16.8 mm. With reference to the cutting step in the turning process, two process parameters, namely the rotary speed and cutting depth were set according to a 3^2 full factorial design. In order to keep the tool wear constant (according to suggestions provided by the tool supplier), we selected the value of the feed rate depending on the level of rotery speed (a feed rate equal to 0.07, 0.11 and 0.14 mm/rev was set when the speed was equal to 80, 70 and 65 m/min, respectively). Table 6.1 shows 9 treatments with different process variables. Each treatment was replicated 10 times. Hence, a set of $9 \times 10 = 90$ samples was used in this experimental study. These two process variables (depth and speed) are recorded in the input matrix $\mathbf{X}$ after the normalization (subtract the mean and divide by standard deviation). After the turning operations, all the 90 machined surfaces were measured with a CMM machine using a touch trigger probe head that holds a four-tip stylus of 0.5 mm radius. More details of the experiment is provided in [122].

The measurements were taken in 42 mm along the bar length direction with 210 cross-sections. Each cross-sections were measured with 64 generatrices. Therefore, a set of 210×64 points, equally distributed on the cylindrical surface, was measured for each sample $\mathbf{Y}_i, i = 1, \cdots, 90$. The examples of the cylindrical surface are shown in Figure 1.5, which clearly shows the shape of the cylinder are influenced by both the rotery speed and cutting depth. Furthermore, the surface roughness are also influenced by these process

Table 6.1: Cutting parameters for 9 experimental conditions

Ex. No	Depth(mm)	Speed(m/min)
1	0.4	80
2	0.4	70
3	0.4	65
4	0.8	80
5	0.8	70
6	0.8	65
7	1.2	80
8	1.2	70
9	1.2	65

variables. To model both the cylindrical mean shape and the residual with unequal variance caused by the different process variables, we combine the framework proposed by [123] with our proposed tensor regression model in the section 6.4.1.

6.4.1 Handling unequal variances of residuals

To model the unequal variances of residuals as a function of the process variables, we assume that the noise $E_i \sim N(0, \sigma_i^2)$, where $\log \sigma_i^2 = \mathbf{x}_i'\boldsymbol{\gamma} + \gamma_0$. Therefore, combining with the tensor regression model in (6.1), the parameter $\boldsymbol{\gamma}$, γ_0 and $\mathcal{A}$ can be estimated by maximizing the likelihood estimation given by $L(\beta, \boldsymbol{\gamma}; y_i) = -\frac{1}{2}(\sum_i I_1 I_2 \log(\sigma_i^2) + \sum_i \frac{\|\mathbf{Y}_i - \bar{\mathbf{Y}} - \mathcal{A} \times_3 \mathbf{X}_i\|^2}{\sigma_i^2})$. The likelihood function can be maximized by iteratively updating $\boldsymbol{\gamma}$ and $\mathcal{A}$ until convergence as follows: 1) For fixed $\boldsymbol{\gamma}$ and γ_0, $\sigma_i^2 = \exp(\mathbf{x}_i'\boldsymbol{\gamma} + \gamma_0)$, with transformation $\mathbf{Y}_i^0 = \frac{\mathbf{Y}_i - \bar{\mathbf{Y}}}{\sigma_i}$, $\mathbf{X}_i^0 = \frac{\mathbf{X}_i}{\sigma_i}$, the MLE can be obtained by the proposed tensor regression methods introduced in section (6.2). 2) For fixed $\mathcal{A}$, MLE becomes the gamma regression with $\log$ link on the Residual Mean Squares Error (RMSE) $\frac{1}{I_1 I_2}\|\hat{\mathbf{E}}_i\|^2$, where $\hat{\mathbf{E}}_i = \mathbf{Y}_i - \mathcal{A} \times_3 \mathbf{X}_i$.

We then apply RTDR on these cylindrical surface to map the relationship of the mean shape and residual variance with process variables. First, the eigentensors of RTDR are extracted, as shown in Figure 6.5. The RMSE and the fitted σ^2 of the 90 samples via the gamma regression are shown in Figure 6.6b. It is clear that the proposed framework

Figure 6.5: Eigen-tensors with regularized Tucker decomposition

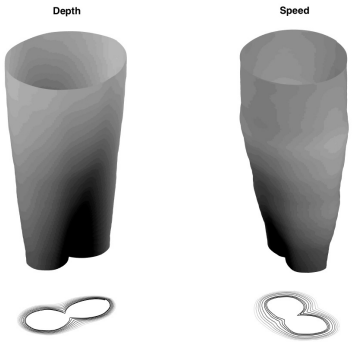
Table 6.2: Gamma regression of $\|\hat{E}_i\|^2$

		Estimate	SE	tStat	pValue
γ_0	Intercept	-13.5755	0.0451	-300.1470	$1e-133$
γ	depth	0.3856	0.0479	8.0525	$1e-12$
	speed	-0.1255	0.0479	-2.6212	0.01

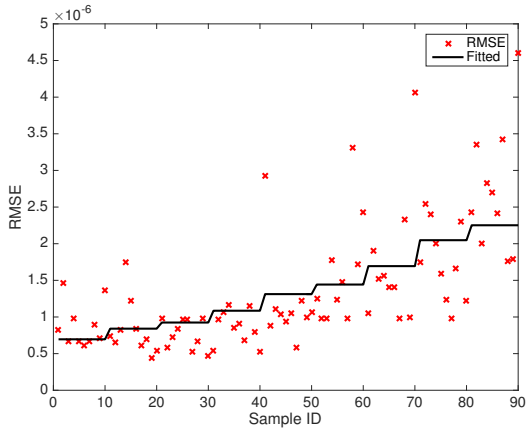
is able to account for the unequal variance under the 9 different input settings. Finally, the gamma regression coefficients of the RMSE are shown in Table 6.2. From this table, we can conclude that if the cutting depth increases or rotery speed decreases, the surface roughness will also increase. Moreover, for the surface roughness, the effect of cutting depth is much more significant than the rotery speed. These findings are consistent with engineer principals.

6.4.2 Process optimization

The estimated tensor regression model can also provide useful information to optimize the process settings (cutting depth and speed) for better product quality. In this turning



(a) Tensor regression coefficient $\mathcal{A}$



(b) Residual Mean of Square Error (RMSE) and fitted $\hat{\sigma}^2$ via gamma regression

Figure 6.6: Result of tensor regression via regularized tucker decomposition

process, the goal is to produce an cylindrical surface with a uniform radius $r_t = 16.8\text{mm}$. Therefore, the following optimization problem can be solved. The objective function is defined as the sum of squared differences of the produced mean shape and the uniform cylinder with radius r_t. Furthermore, we require the produced surface roughness σ to be smaller than a certain threshold σ_0. Finally, the process variable are typically constrained in certain range $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$ due to the physical constraints of the machine.

$$\min_{\mathbf{x}} \|\bar{\mathbf{Y}} + \hat{\mathcal{A}} \times_3 \mathbf{x} - r_t\|_F^2 \quad s.t. \sigma \leq \sigma_0, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$$

It is straightforward to show that this optimization problem can be reformulated to a Quadratic Programming (QP) model with linear constraints as

$$\min_{\mathbf{x}} \mathbf{x}^T \mathbf{A}_{(3)}^T \mathbf{A}_{(3)} \mathbf{x} + 2\mathbf{x}^T \mathbf{A}_{(3)}^T (\text{vec}(\bar{\mathbf{Y}}) - r_t) \quad s.t. \boldsymbol{\gamma}' \mathbf{x} \leq \log(\sigma_0^2) - \gamma_0, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$$

Since the problem is convex, it can be solved via the standard Quadratic Programming approach. For example, if we constrain $\sigma_0 = 0.0001$ and process variables lies in the range of the design Table 6.1. The optimal rotery speed can be solved as $80\text{m}/\text{min}$ and the optimal cutting depth as 0.8250mm. Under this setting, we simulate the produced

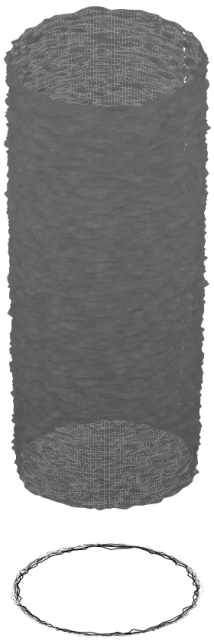


Figure 6.7: Simulated cylinder under the optimal setting (rotery speed: $80\mathrm{m}/\mathrm{min}$, cutting depth: $0.8250\mathrm{mm}$)

cylindrical surfaces as shown in Figure 6.7 by combining both the predicted surface $\hat{\mathbf{Y}} = \bar{\mathbf{Y}} + \hat{\mathcal{A}} \times_3 \mathbf{x}$ and added noise from the normal distribution with the estimated standard deviation $\hat{\sigma} = \exp(\frac{1}{2}(\gamma_0 + \boldsymbol{\gamma}'\hat{\mathbf{x}}))$. It is clear that the produced cylindrical surfaces under this optimal setting is closer to the uniform cylinder compared to other input settings as shown in Figure 1.5.

6.5 Conclusion

Point cloud modeling is an important research area with various applications especially in modern manufacturing due to the popularity of 3D scanning tools and the need for accurate shape control. As most structured point clouds can be represented in the tensor format in certain coordinate system, in this paper, we proposed two tensor regression strategies (RTR and RTDR) to link the response tensor with input variables. In the simulation study, we showed the proposed methods outperform the existing vector-based techniques. Finally, the proposed methods were applied to a real case study on the point cloud modeling in a turning process. The results indicated that our methods are capable of handling residuals

with unequal variances due to the different roughness caused by different input settings. We also demonstrated that how this model can be used to find the optimal setting of the process variables.

CHAPTER 7
CONCLUSION

7.1 Summary of Original Contributions

In summary, this thesis investigates four major research topics in the area of high-dimensional functional data analysis. In Chapter 3, we consdier detecting anomaly from HD functional profiles or single images. This method can be applied to various inspection system or online image based monitoring system. In Chapter 4, we extend this framwork into HD functional streaming data with complex spatial temporal structure. In Chapter 5, we still focus on the anomaly detection but weconsider the case where we don't have the full sampling. Therefore, an adpative sampling strategy is needed for online anomaly detection. In Chapter 6, we study a new topic, which model the complex 3D shape with the process variables via regualrized tensor regression.

The original contribution of Chapter 3 is to proposed a novel methodology for anomaly detection in noisy images with smooth backgrounds. The proposed method, named smooth-sparse decomposition, exploits regularized high-dimensional regression to decompose an image and separate anomalous regions by solving a large-scale optimization problem. This one-step approach is much more efficient than the existing first-smooth-then-detect approaches. To enable the proposed method for real-time implementation, a fast algorithm for solving the optimization model is proposed. Using simulations and a case study, we evaluate the performance of the proposed method and compare it with existing methods. Numerical results demonstrate the superiority of the proposed method in terms of the detection accuracy as well as computation time. This methodology has great potential impacts in the image based inspection methods and it actually has been tested on various inspection methods such as Guided Wave Test, Photo-elasticity, Thermal Imaging, Rolling Inspection,

etc.

The original contribution of Chapter 4 is to propose a novel process monitoring methodology for high-dimensional data streams including profiles and images that can effectively address foregoing challenges. We introduce spatio-temporal smooth sparse decomposition (ST-SSD), which serves as a dimension reduction and denoising technique by decomposing the original tensor into the functional mean, sparse anomalies, and random noises. ST-SSD is followed by a sequential likelihood ratio test on extracted anomalies for process monitoring. To enable real-time implementation of the proposed methodology, recursive estimation procedures for ST-SSD are developed. ST-SSD also provides useful diagnostics information about the location of change in the functional mean. We also use real-world case studies to show that all the aforementioned methodologies can be implemented in multichannel signals, as well as video streams. The studies include online rolling bar inspection, online solar flare monitoring, and forging signal monitoring. Currently, we are also working with our industrial collaborators to implement the methodologies in their monitoring and inspection systems.

The original contribution of Chapter 5 is to develop a novel framework named Adaptive Kernelized Maximum-Minimum Distance $(\mathrm{AKM^2D})$ to speed up the inspection and anomaly detection process through an intelligent sequential sampling scheme integrated with fast estimation and detection. The proposed method balances the sampling efforts between the space filling sampling (exploration) and focused sampling near the anomalous region (exploitation). The proposed methodology is validated by conducting simulations and a case study of anomaly detection in composite sheets using a guided wave test. I have also been working with researchers from Aerospace Engineering to design compressive sensing framework to reconstruct HD data with partial sampling. We are currently working on combining my adaptive sensing method with this compressive sensing framework to further reduce the measurement time. We have shown that this innovative combination can reduce the data acquisition time of point measurement systems from 4 hours to several

minutes without losing the detection powers.

The original contribution of Chapter 6 is to propose regularized tucker decomposition and regularized tensor regression to model the variational patterns of point clouds and link them to process variables. The performance of the proposed method is evaluated through simulation studies. We also applied this framework to the Turning process to model the 3D shape variations of the product. This model can also help to determine the best parametric settings for the precision control of the product.

7.2 Future Work

The problems of high-dimensional data analysis for system monitoring, anomaly detection, and system evaluation is a very active research currently and there are many future step.

The possible extension of the Chapter 3 and Chapter 4 is generalize SSD for other types of spatial and temporal structures such as non-smooth and/or periodic functional mean. To model different types of spatial or temporal structures, one may adjust the basis for example by using Fourier or wavelet basis. Another non-trival generation is to propose a data-adaptive method to learn the "best" basis to model the anomaly components. Similar in Chapter 6, there are several potential research directions to be investigated. One possible extension is to extend this method to non-smooth point cloud with abrupt changes in surface. Another extension is to extend this method to unstructured point cloud.

For other possible directions, combining the HD data analysis and parallel processing to address the challenges of both large sample size (e.g. the big data challenge) and high-dimensionality is very important. For example, the manufacturing process can provide millions of samples in the production lines with HD data measured from thousands of stations. Therefore, when the number of samples becomes large, traditional statistical methods with complexity or even may not scale well and cannot be implemented to address those problems in real-time. Therefore, designing scalable computational algorithms with less complexity is needed to address the big-data challenge.

Appendices

APPENDIX A

APPENDIX ON "ANOMALY DETECTION FOR IMAGES AND HIGH-DIMENSIONAL SPATIAL FUNCTIONAL PROFILE"

Appendix A

Proposition. *If B_a is orthogonal, in iteration k, the subproblem $\hat{\theta}_a^{(k)} = \text{argmin}_{\theta_S} \|y - B\theta^{(k)} - B_a\theta_a\|^2 + \gamma\|\theta_a\|_1$ has a closed-form solution in the form of $\hat{\theta}_a^{(k)} = S_{\frac{\gamma}{2}}(B_a^T(y - B\theta^{(k)}))$, in which $S_\gamma(x) = \text{sgn}(x)(|x| - \gamma)_+$ is the soft-thresholding operator, and $\text{sgn}(x)$ is the sign function and $x_+ = \max(x, 0)$.*

Proof. If B_a is orthogonal, in each iteration k, we solve $\theta_S^{(k)} = \text{argmin}_{\theta_S} \|y - B\theta^{(k)} - B_a\theta_a\|^2 + \gamma\|\theta_a\|_1$. The first Karush–Kuhn–Tucker (KKT) condition of this optimization problem can be expressed as: $\nabla\|y - B\theta^{(k)} - B_a\theta_a\|^2 + \gamma g = 0$, where ∇ is the gradient operator and $g = [g_i] = \begin{cases} \text{sgn}(\theta_{Si}) & \theta_{Si} \neq 0 \\ [-1, 1] & \theta_{Si} = 0 \end{cases}$. The square is $\|y - B\theta^{(k)} - B_a\theta_a\|^2 = \theta_a^T B_a^T B_a \theta_a - 2\theta_a^T B_a^T (y - B\theta^{(k)}) + \|y - B\theta^{(k)}\|^2$. Since $B_a^T B_a = I$, the loss function can be simplified to $\|y - B\theta^{(k)} - B_a\theta_a\|^2 = \theta_a^T\theta_a - 2\theta_a^T B_a^T(y - B\theta^{(k)}) + \|y - B\theta^{(k)}\|^2$. Consequently, after simplification, the KKT condition gives $\theta_a = B_a^T(y - B\theta^{(k)}) - \frac{\gamma}{2}g$. We consider two cases for this solution, if $\theta_{ai} \neq 0$, then $\theta_{ai} + \frac{\gamma}{2}\text{sgn}(\theta_{ai}) = B_S^T(y - B\theta^{(k)})$. If $\theta_{ai} = 0$, then $B_a^T(y - B\theta^{(k)}) = \frac{\gamma}{2}g \in [-\frac{\gamma}{2}, \frac{\gamma}{2}]$. The solution can be given in a compact form of $\theta_a^{(k+1)} = \text{sgn}(B_a^T(y - B\theta^{(k)}))(\left|B_a^T(y - B\theta^{(k)})\right| - \frac{\gamma}{2})_+$prop* , which is a soft-thresholding operator denoted by $S_{\frac{\gamma}{2}}(B_a^T(y - B\theta^{(k)}))$. □

Appendix B

Proposition. *The BCD algorithm attains the global optimum of the SSD loss function in (A.1).*

$$\underset{\theta,\theta_a}{argmin}\, \|e\|^2 + \lambda\theta^T R\theta + \gamma|\theta_a|_1, \;\; \text{subject to. } y = B\theta + B_a\theta_a + e \tag{A.1}$$

Proof. [124] in page 484, Theorem 5.1 proved that if an objective function f can be decomposed into the sum of a continuous function f_0 and some non-differentiable functions f_i $= 1, \cdots, N$, with some basic continuity assumptions on f_0, the BCD algorithm guarantees to attain a local optimum. It is clear that the SSD objective function in (A.1) is comprised of a continuous function $\|e\|^2 + \lambda\theta' R\theta$ and a non-differentiable penalty term $\gamma|\theta_S|_1$. Consequently, the BCD algorithm converges to a local optimum. In addition, since problem (A.1) is convex, the attained optimum is the global optimum. □

Appendix C:

Proposition. *The SSD problem in (A.1) is equivalent to a weighted LASSO problem in the form of*

$$\arg\min_{\theta_S} F(\theta_S) = (y - B_a\theta_a)^T(I - H)(y - B_a\theta_a) + \gamma\|\theta_a\|_1 \tag{A.2}$$

with $H = B(B^TB + \lambda R)^{-1}B^T$.

Proof. We first solve (A.1) for θ by fixing θ_S. That is $\theta = argmin_\theta \|y - B\theta - B_a\theta_a\|^2 + \lambda\theta^T R\theta + \gamma\|\theta_a\|_1$, which can be solved via $\theta = (B^TB + \lambda R)^{-1}B^T(y - B_a\theta_a)$. Thus, it can be written that $B\theta = B(B^TB + \lambda R)^{-1}B^T(y - B_a\theta_a) = H(y - B_a\theta_a)$. By plugging in this into (A.1), we have $\theta = argmin_\theta \|y - H(y - B_a\theta_a) - B_a\theta_a\|^2 + \lambda(y - B_a\theta_a)^T H^T RH(y - B_a\theta_a) + \gamma\|\theta_S\|_1$. After simplification and since $(I - H)^2 + \lambda BK_\lambda^{-1}RK_\lambda^{-1}B^T(y - B_a\theta_a) = I - H$, where $K_\lambda = B^TB + \lambda R$, we can show that $\|y - B\theta - B_a\theta_a\|^2 + \lambda\theta^T R\theta + \gamma\|\theta_a\|_1 = (y - B_a\theta_a)^T(I - H)(y - B_a\theta_a) + \gamma\|\theta_a\|_1$, which is the weighted LASSO formulation. □

Appendix D:

Claim 1. *The* $f(\theta_a) = (y - B_a\theta_a)^T(I - H)(y - B_a\theta_a)$. *is convex for* θ_a.

Proof. $f(\theta_a) = (y - B_a\theta_a)^T(I - H)(y - B_a\theta_a)$. To prove $f(\theta_a)$ is convex, we only need to show that $I - H$ is a positive semi-definite matrix. From Appendix C, it is given that $I-H = (I-H)^2+\lambda B(B^TB+\lambda R)^{-1}R(B^TB+\lambda R)^{-1}B^T$. Clearly, the first term $(I-H)^2$ is a positive semi-definite matrix. For the second term, since R is a positive semi-definite matrix, $B(B^TB+\lambda R)^{-1}R(B^TB+\lambda R)^{-1}B^T$ is also positive semi-definite. Consequently, $f(\theta_a)$ is a convex function. □

Appendix E:

Claim 2. *$f(\cdot)$ is Lipschitz continuos, in which satisfies $\|\nabla f(a) - \nabla f(b)\| \leq L\|a - b\|$ for any $a, b \in R$ with $L = 2\|B_a\|_2^2$*

Proof. We first show that H is positive semidefinite matrix. $H = B(B^TB + \lambda R)^{-1}B^T$. Since $B^TB+\lambda R$ is positive definite matrix, $(B^TB+\lambda R)^{-1}$ is also positive definite matrix, and H is positive semi-definite matrix.

We then prove that $\|I - H\|_2 \leq 1$. Notice that $\|X\|_2$ refers to the spectrum norm of matrix X. This is because that $\|I - H\|_2 = \sqrt{\lambda_{max}[(I - H)^2]} = \lambda_{max}(I - H) = 1 - \lambda_{min}(H) \leq 1$. The last equation hold because $\lambda_{min}(H) \geq 0$ since H is positive semi-definite matrix. Note that $\lambda_{max}(X)$ refers to the largest eigenvalue of X and $\lambda_{min}(X)$ refers to the smallest eigenvalue of X.

Consequently, $\nabla f(a) = \nabla(y - B_a a)^T(I - H)(y - B_a a) = 2B_a^T(I - H)(B_a a - y)$. $\|\nabla f(a)-\nabla f(b)\| = \|2B_a^T(I-H)B_a(a-b)\| \leq \|2B_a^T(I-H)B_a\|_2 \cdot \|a-b\| \leq L\|a-b\|$, in which $L = 2\|B_a\|_2^2$

The last equation holds because $\|2B_a^T(I - H)B_a\|_2 \leq \|2B_a^T\|_2\|(I - H)\|_2\|B_a\|_2 \leq \|2B_a^T\|_2\|B_S\|_2 = 2\|B_a\|_2^2$. □

Appendix F:

Proposition. *The proximal gradient method for the SSD problem in (A.1), given by* $\theta_a^{(k)} = argmin_{\theta_a}\{f(\theta_a^{(k-1)}) + \left\langle \theta_a - \theta_a^{(k-1)}, \nabla f(\theta_a^{(k-1)}) \right\rangle + \frac{L}{2}\|\theta_a - \theta_a^{(k-1)}\|^2 + \gamma\|\theta_a\|_1\}$*, has a closed-form solution in each iteration* k*, in the form of a soft-thresholding function as follows:*

$$\theta_a^{(k)} = S_{\frac{\gamma}{L}}(\theta_a^{(k-1)} + \frac{2}{L}B_a^T(y - B_a\theta_a^{(k-1)} - \mu^{(k)})) \tag{A.3}$$

with $L = 2\|B_a\|_2^2$.

Proof. Since $\nabla f(\theta_a^{(k-1)}) = 2B_a^T B_a\theta_a^{(k-1)} - 2B_a^T(y - B\theta^{(k)})$, in each iteration given $\theta_a^{(k-1)}$, the $\theta_a^{(k)} = argmin_{\theta_a}\|\theta_a - \theta_a^{(k-1)} - \frac{2}{L}B_a^T(y - B\theta^{(k)} - B_a\theta_a^{(k-1)})\|^2 + \gamma\|\theta_a\|_1\}\}$. Thus, similar to Appendix A, it is simple to show that this problem can be solved using a soft thresholding operator in the form of $\theta_a^{(k)} = S_{\frac{\gamma}{L}}(\theta_a^{(k-1)} + \frac{2}{L}B_a^T(y - B_a\theta_a^{(k-1)} - \mu^{(k)}))$. □

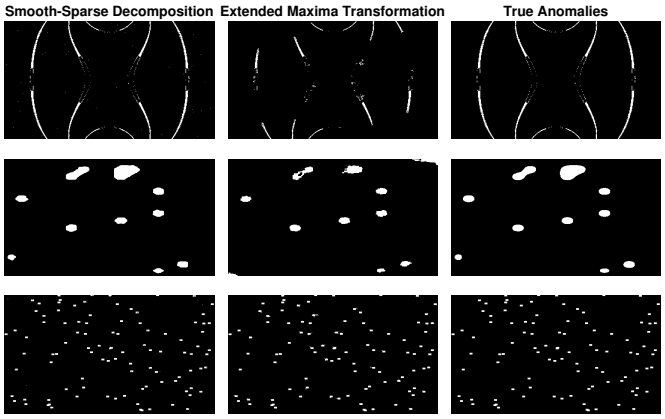
Appendix G:

Claim 3. *Suppose the Cholesky decomposition of* $B_i^T B_i$ *is given as* $B_i^T B_i = Z_i Z_i^T$*, the eigen decomposition* $Z_i^{-1}D_i^T D_i(Z_i^{-1})^T$ *is* $U_i\mathrm{diag}(s_i)U_i^T$ *and* $A_i = B_i(Z_i^{-1})^T U_i$*. It can be shown that* $H_i(\lambda) = A_i^T\mathrm{diag}(\frac{1}{1+\lambda s_1}, \cdots, \frac{1}{1+\lambda s_n})A_i$*, and its trace is given by* $\mathrm{tr}(H_i) = \sum_{i=1}^n \frac{1}{1+\lambda s_i}$

Proof. The proof of the first part is given below:

$$\begin{aligned}
H_i(\lambda) &= B_i(B_i^T B_i + \lambda D_i^T D_i)^{-1}B_i^T = B_i(Z_i Z_i^T + \lambda D_i^T D_i)^{-1}B_i^T \\
&= B_i(Z_i^{-1})^T(I + \lambda Z_i^{-1}D_i^T D_i(Z_i^{-1})^T)^{-1}(Z_i^{-1})B_i^T \\
&= B_i(Z_i^{-1})^T(I + \lambda U_i\mathrm{diag(s_i)}U_i^T)^{-1}(Z_i^{-1})B_i^T \\
&= B_i(Z_i^{-1})^T U_i(I + \lambda\mathrm{diag(s_i)})^{-1}U_i^T(Z_i^{-1})B_i^T \\
&= A_i(I + \lambda\mathrm{diag(s_i)})^{-1}A_i^T \\
&= A_i^T\mathrm{diag}(\frac{1}{1+\lambda s_1}, \cdots, \frac{1}{1+\lambda s_n})A_i
\end{aligned}$$

Figure A.1: Anomalies detection comparison result for SSD and extended maxima transformation when $\delta = 0.3$



To compute the trace of H_i, we first show that $A_i^T A_i = U_i^T Z_i^{-1} B_i^T B_i (Z_i^{-1})^T U_i = U_i^T U_i = I$. Thus the trace of H_i becomes $\mathrm{tr}(H_i) = \mathrm{tr}(A_i(I + \lambda \mathrm{diag(s_i)})^{-1} A_i^T) = \mathrm{tr}(A_i^T A_i (I + \lambda \mathrm{diag(s_i)})^{-1}) = \mathrm{tr}((I + \lambda \mathrm{diag(s_i)})^{-1}) = \sum_{i=1}^{n} \frac{1}{1+\lambda s_i}$ □

Appendix H:

"In this appendix, we applied the extended-maxima transformationmethod to the simulated images with line anomalies, clustered anomalies and scattered anomalies. The detection results are reportedin Figure A.1. Moreover, the FPR, FNR, and computational time for all the benchmark methods are reported in Table A.1. "

Table A.1: FPR, FNR, and computation time for line , clustered and scattered anomalies with $\delta = 0.1, 0.2, 0.3$

δ	Defect Type	Criterion	SSD	Edge	Jump	Local	Global	Maxima
0.1	Line	FPR	0.108	0.012	0.022	0.066	0.202	0.045
		FNR	0.234	0.989	0.908	0.492	0.591	0.791
	Clustered	FPR	0.016	0.0003	0.086	0.539	0.211	0.008
		FNR	0.035	0.979	0.837	0.756	0.799	0.868
	Scattered	FPR	0.011	0.008	0.179	0.019	0.204	0.018
		FNR	0.076	0.858	0.722	0.567	0.752	0.984
0.2	Line	FPR	0.027	0.016	0.037	0.058	0.202	0.005
		FNR	0.021	0.900	0.126	0.181	0.507	0.792
	Clustered	FPR	0.017	0.0003	0.083	0.052	0.213	0.002
		FNR	0.005	0.89	0.127	0.462	0.673	0.657
	Scattered	FPR	0.0114	0.005	0.138	0.02	0.203	0.004
		FNR	0.0153	0.293	0.108	0.251	0.595	0.038
0.3	Line	FPR	0.001	0.015	0.035	0.054	0.195	0.001
		FNR	0.003	0.783	0.111	0.063	0.456	0.557
	Clustered	FPR	0.018	0.001	0.081	0.046	0.211	0.007
		FNR	0.001	0.754	0.054	0.289	0.572	0.268
	Scattered	FPR	0.012	0.003	0.11	0.02	0.203	0.001
		FNR	0.007	0.257	0.063	0.087	0.407	0.012
Computational Time			0.19s	0.667s	38.43s	0.043s	0.048s	0.039s

'SSD' for Smooth Sparse Decomposition, 'Edge' for edge detection, 'Jump' for jump regression, 'Local' for local thresholding, 'Global' for global thresholding, and 'Maxima' for extented maxima transformation.

APPENDIX B

APPENDIX ON "ANOMALY DETECTION FOR IMAGES AND HIGH-DIMENSIONAL SPATIAL FUNCTIONAL PROFILE"

Appendix A: Decomposition of the projection matrix Since $B_s^T B_s + R_s = \otimes_{i=l}^{1}(B_{si}^T B_{si} + R_{si})$, from the property of Kronecker product, we know that $(B_s^T B_s + R_s)^{-1} = \otimes_{i=l}^{1}(B_{si}^T B_{si} + R_{si})^{-1}$.

Finally, we have $H_s = B_s(B_s^T B_s + R_s)^{-1} B_s^T = \otimes_{i=l}^{1} B_{si}(B_{si}^T B_{si} + R_{si})^{-1} B_{si}^T = \otimes_{i=l}^{1} H_{si}$

Appendix B: Prove of the recursive estimation of H_t We can apply the standard block matrix inversion formula as follow, $M = \begin{bmatrix} A & b \\ b^T & d \end{bmatrix} \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{(n-1)\times(n-1)}$, $b \in \mathbb{R}^{(n-1)\times 1}$, g is a scalar, then $M^{-1} = \begin{bmatrix} A^{-1}(I + bb^T A^{-1} g) & -A^{-1} bg \\ -b^T A^{-1} g & g \end{bmatrix}$, with $g = (d - b^T A^{-1} b)^{-1}$.

Therefore, $K_{\lambda_t,t} = (K_t + \lambda_t I)^{-1} = \begin{bmatrix} K_{t-1} + \lambda_t I_{t-1} & k_{t-1} \\ k_{t-1}^T & 1 + \lambda_t \end{bmatrix}^{-1}$. Following this , it is straightforward to show that $K_{\lambda_t,t} = \begin{bmatrix} K_{\lambda_t,t-1}(I + k_{t-1} k_{t-1}^T K_{\lambda_t,t-1} g_{t-1}) & -K_{\lambda_t,t-1} k_{t-1} g_{t-1} \\ -k_{t-1}^T K_{\lambda_t,t-1} g_{t-1} & g_{t-1} \end{bmatrix}$, where $g_{t-1} = (1 + \lambda_t - r_{t-1}^T k_{t-1})^{-1}$. Therefore, the $\tilde{H}_t$ can be computed recursively by

$$\begin{aligned} \tilde{H}_t &= K_t K_{\lambda_t,t} = \begin{bmatrix} K_{t-1} & k_{t-1}^T \\ k_{t-1} & 1 \end{bmatrix} \begin{bmatrix} K_{\lambda_t,t-1}(I + k_{t-1} k_{t-1}^T K_{\lambda_t,t-1} g_{t-1}) & -K_{\lambda_t,t-1} k_{t-1} g_{t-1} \\ -k_{t-1}^T K_{\lambda_t,t-1} g_{t-1} & g_{t-1} \end{bmatrix} \\ &= \begin{bmatrix} H_{t-1} - k_{t-1} r_{t-1}^T g_{t-1}(I_{t-1} - H_{t-1}) & (I_{t-1} - H_{t-1}) k_{t-1} g_{t-1} \\ r_{t-1}^T (I_{t-1} + k_{t-1} r_{t-1} g_{t-1} - g_{t-1}) & (1 - r_{t-1}^T k_{t-1}) g_{t-1} \end{bmatrix} \end{aligned}$$

where $r_t = K_{\lambda_t,t} k_t$.

Appendix C: Roughness minimization modeling estimator $\hat{\theta}_t$

Proof. First, from the property of Kronecker product we know that $\theta^T(A \otimes B)\theta = tr(A\Theta^T B\Theta)$ if $\theta = \mathrm{vec}(\Theta)$. The penalization term can be reduced to

$$\begin{aligned}
\theta^T R\theta &= \theta^T(I_t \otimes R_s + \lambda_t D_t^T D_t \otimes B_s^T B_s + \lambda_t D_t^T D_t \otimes R_s)\theta \\
&= tr(\Theta^T R_s \Theta + D_t \Theta^T (B_s^T B_s + R_s)\Theta D_t^T) \\
&= \sum_{i=1}^{t} (\theta_i R_s \theta_i + (\theta_{i+1} - \theta_i)^T (B_s^T B_s + R_s)(\theta_{i+1} - \theta_i))
\end{aligned}$$

Finally $\hat{\theta}_t$ can be solved by

$$\begin{aligned}
\hat{\theta}_t &= \underset{\theta_t}{\mathrm{argmin}} \sum_{i=1}^{t} ((\theta_i R_s \theta_i + (\theta_{i+1} - \theta_i)^T (B_s^T B_s + R_s)(\theta_{i+1} - \theta_i)) + \|y_t - B_s\theta_t - a_t\|^2) \\
&= \underset{\theta_t}{\mathrm{argmin}}\, \lambda_t(\theta_t - \theta_{t-1})^T (B_s^T B_s + R_s)(\theta_t - \theta_{t-1}) + \theta_t^T R_s \theta_t + \|y_t - B_s\theta_t - a_t\|^2 \quad \text{(B.1)} \\
&= \underset{\theta_t}{\mathrm{argmin}} (1 + \lambda_t)\theta_t^T (B_s^T B_s + R_s)\theta_t - 2\theta_t^T (\lambda_t B_s^T B_s \theta_{t-1} + \lambda_t R_s \theta_{t-1} + B_s^T (y_t - S_t)) \\
&= (\frac{\lambda_t}{1 + \lambda_t}\theta_{t-1} + \frac{1}{1 + \lambda_t}(B_s^T B_s + R)^{-1} B_s^T (y_t - a_t)) \\
&= (1 - \tilde{\lambda}_t)\theta_{t-1} + \tilde{\lambda}_t (B_s^T B_s + R)^{-1} B_s^T (y_t - a_t)
\end{aligned}$$

The first equation holds since $\theta_1, \cdots, \theta_{t-1}$ is fixed, only the last term of the summation ($i = t$) is considered. Finally, we know that $\hat{y}_t = B_s\theta_t = (1 - \tilde{\lambda}_t)\hat{y}_{t-1} + \tilde{\lambda}_t H_s(y_t - a_t)$ because $H_s = B_s(B_s^T B_s + R_s)^{-1} B_s^T$. □

Appendix D: Equivalency of Equation (4.8) to weighted lasso formulation

Proof. According to Appendix A, we have solved θ_t by fixing other variables as $\hat{\theta}_t = \frac{\lambda_t}{1+\lambda_t}\theta_{t-1} + \frac{1}{1+\lambda_t}(B_s^T B_s + R)^{-1} B_s^T (y_t - a_t)$. Then, by plugging it into (B.1), and considering the terms that only contain $y_t - a_t$, we have

$$\lambda_t(\theta_t - \theta_{t-1})^T (B_s^T B_s + R_s)(\theta_t - \theta_{t-1}) = \tilde{\lambda}_t(1 - \tilde{\lambda}_t)((y_t - a_t)^T B_s (B_s^T B_s + R)^{-1} -$$

$\theta_{t-1}^T)(B_s^T B_s + R_s)((B_s^T B_s + R)^{-1} B_s^T (y_t - a_t) - \theta_{t-1}) = \tilde{\lambda}_t(1 - \tilde{\lambda}_t)((y_t - a_t)^T H_s(y_t - a_t) - 2(y_t - a_t)^T B_s \theta_{t-1}) + C_0$

$\theta_t^T R_s \theta_t = \tilde{\lambda}_t^2 (y_t - a_t)^T B_s (B_s^T B_s + R_s)^{-1} R_s (B_s^T B_s + R)^{-1} B_s^T (y_t - a_t) + 2\tilde{\lambda}_t(1 - \tilde{\lambda}_t)(y_t - a_t)^T B_s (B_s^T B_s + R_s)^{-1} R_s \theta_{t-1} + C_1$

$\|y_t - B_s\theta_t - a_t\|^2 = \|(I - \tilde{\lambda}_t H_s)(y_t - a_t) - (1 - \tilde{\lambda}_t)\hat{y}_{t-1}\|^2 = (y_t - a_t)^T (I - \tilde{\lambda}_t H_s)^2 (y_t - a_t) - 2(1 - \tilde{\lambda}_t)(y_t - a_t)^T (I - \tilde{\lambda}_t H_s)\hat{y}_{t-1} + C_2$

C_0, C_1, C_2 are the constant terms that do not include a_t. Finally, by only taking consideration of the quadratic and linear term of $y_t - a_t$. Equation (4.8) becomes:

$$\begin{aligned} & \|y_t - B_s\theta_t - a_t\|^2 + \lambda_t(\theta_t - \theta_{t-1})^T (B_s^T B_s + R_s)(\theta_t - \theta_{t-1}) + \theta_t^T R_s \theta_t + \gamma \|\theta_{a,t}\|_1 \\ = \; & (y_t - a_t)^T Q (y_t - a_t) + (y_t - a_t)^T P + \gamma \|\theta_{a,t}\|_1 \end{aligned} \tag{B.2}$$

In which

$$\begin{aligned} Q &= \tilde{\lambda}_t(1 - \tilde{\lambda}_t) H_s + \tilde{\lambda}_t^2 B_s (B_s^T B_s + R_s)^{-1} R_s (B_s^T B_s + R)^{-1} B_s^T + (I - \tilde{\lambda}_t H_s)^2 \\ &= (\tilde{\lambda}_t H_s - \tilde{\lambda}_t^2 H_s) + \tilde{\lambda}_t^2 (H_s - H_s^2) + I - 2\tilde{\lambda}_t H_s + \tilde{\lambda}_t^2 H_s^2 \\ &= I - \tilde{\lambda}_t H_s \end{aligned}$$

The second '=' holds because $B_s(B_s^T B_s + R_s)^{-1} R_s (B_s^T B_s + R)^{-1} B_s^T = H_s - H_s^2$ and

$$\begin{aligned} P &= 2\tilde{\lambda}_t(1 - \tilde{\lambda}_t) B_s (B_s^T B_s + R_s)^{-1} R_s \theta_{t-1} + 2B_s\theta_{t-1} - 2(1 - \tilde{\lambda}_t)(I - \tilde{\lambda}_t H_s)\hat{y}_{t-1} \\ &= 2\tilde{\lambda}_t(1 - \tilde{\lambda}_t) B_s ((B_s^T B_s + R_s)^{-1} R_s - I)\theta_{t-1} - 2(1 - \tilde{\lambda}_t)(I - \tilde{\lambda}_t H_s)\hat{y}_{t-1} \\ &= -2\tilde{\lambda}_t(1 - \tilde{\lambda}_t) H_s B_s \hat{y}_{t-1} - 2(1 - \tilde{\lambda}_t)(I - \tilde{\lambda}_t H_s)\hat{y}_{t-1} \\ &= -2(1 - \tilde{\lambda}_t)\hat{y}_{t-1} \end{aligned}$$

The third '=' holds because $B_s((B_s^T B_s + R_s)^{-1} R_s - I)\theta_{t-1} = -B_s(B_s^T B_s + R_s)^{-1} B_s^T B_s \theta_{t-1} =$

$-H_s B_s \theta_{t-1} = -H_s \hat{y}_{t-1}$

Finally, plugging P, Q and $a_t = B_{as}\theta_{a,t}$ into (B.2), we will have (A.2). □

Appendix E: Convexity of $f(\theta_a) = (y_t - B_{as}\theta_{a,t})^T(I - \tilde{\lambda}_t H_s)(y_t - B_{as}\theta_{a,t}) - 2(1-\tilde{\lambda}_t)(y_t - B_{as}\theta_{a,t})^T y_{t-1}$

To prove $f(\theta_a)$ is convex, we only need to show that $I - \tilde{\lambda}_t H_s$ is a positive semi-definite matrix, in which $\tilde{\lambda}_t = \frac{1}{1+\lambda_t} \in (0, 1)$.

We first show that H_s is positive semi-definite matrix. $H_s = B_s(B_s^T B_s + \lambda_s R_s)^{-1} B_s^T$. Since $(B_s^T B_s + \lambda_s R_s)^{-1}$ is a positive definite matrix, we know H_s is also a positive definite matrix.

We then show that $I - H_s$ is positive semi-definite matrix by $I - H_s = (I - H_s)^2 + \tilde{\lambda}_t B_s(B_s^T B_s + \lambda_s R_s)^{-1} R_s (B_s^T B_s + \lambda_s R_s)^{-1} B_s^T$, and both terms are positive semi-definite matrices.

We then know $I - \tilde{\lambda}_t H_s = \tilde{\lambda}_t(I - H_s) + (1 - \tilde{\lambda}_t)I$ is also a positive definite matrix.

Appendix F: Lipschitz continuity of $f(\cdot)$ $f(\cdot)$ satisfies$\|\nabla f(\alpha) - \nabla f(\beta)\| \leq L\|\alpha - \beta\|$ for any $\alpha, \beta \in R$ with $L = 2\|B_{as}\|_2^2$

We first proved that $\|I - \tilde{\lambda}_t H_s\|_2 \leq 1$. Notice that $\|X\|_2$ refers to the spectrum norm of matrix X. From the definition of the spectrum norm, we know that $\|X\|_2 = \sqrt{\lambda_{max}(X^T X)}$. Consequently, $\|I - \tilde{\lambda}_t H\|_2 = \sqrt{\lambda_{max}[(I - \tilde{\lambda}_t H)^2]} = \lambda_{max}(I - \tilde{\lambda}_t H) = 1 - \lambda_{min}(\tilde{\lambda}_t H) \leq 1$. For any $\tilde{\lambda}_t \in (0, 1)$.

We then know from Appendix D that $\nabla f(\alpha) = -2B_{as}^T(I - \tilde{\lambda}_t H_s)(y_t - B_{as}\alpha) + 2(1 - \tilde{\lambda}_t)B_{as}^T y_{t-1}$ and

$\|\nabla f(\alpha) - \nabla f(\beta)\| = \|2B_{as}^T(I - \tilde{\lambda}_t H_s)B_{as}(\alpha - \beta)\| \leq \|2B_{as}^T(I - \tilde{\lambda}_t H_s)B_{as}\|_2 \cdot \|\alpha - \beta\| \leq L\|\alpha - \beta\|$, in which $L = 2\|B_{as}\|_2^2$. The last equation holds because $\|2B_{as}^T(I - \tilde{\lambda}_t H_s)B_{as}\|_2 \leq \|2B_{as}^T\|_2 \|I - \tilde{\lambda}_t H_s\|_2 \|B_{as}\|_2 \leq \|2B_{as}^T\|_2 \|B_{as}\|_2 = 2\|B_{as}\|_2^2$.

Appendix G: Solution of $\theta_{a,t}^{(k)}$ **in proximal gradient algorithm** It is not hard to show that the proximal gradient method for (A.2) given by

$$\theta_{a,t}^{(k)} = \underset{\theta_{a,t}}{\operatorname{argmin}}\{f(\theta_{a,t}^{(k-1)}) + \left\langle \theta_{a,t} - \theta_{a,t}^{(k-1)}, \nabla f(\theta_{a,t}^{(k-1)}) \right\rangle + \frac{L}{2}\|\theta_{a,t} - \theta_{a,t}^{(k-1)}\|^2 + \gamma\|\theta_{a,t}\|_1\}$$

has a closed-form solution in each iteration k and can be solved. Since $f(\theta_a) = (y_t - B_{as}\theta_{a,t})^T(I - \tilde{\lambda}_t H_s)(y_t - B_{as}\theta_{a,t}) - 2(1 - \tilde{\lambda}_t)(y_t - B_{as}\theta_{a,t})^T y_{t-1}$

We know that

$$\begin{aligned}
\nabla f(\theta_{a,t}^{(k-1)}) &= -2B_{as}^T(I - \tilde{\lambda}_t H_s)(y_t - B_{as}\theta_{a,t}^{(k-1)}) + 2(1 - \tilde{\lambda}_t)B_{as}^T y_{t-1} \\
&= -2B_{as}^T(y_t - B_{as}\theta_{a,t}^{(k-1)}) + 2B_{as}^T((1 - \tilde{\lambda}_t)y_{t-1} + \tilde{\lambda}_t H_s(y_t - B_{as}\theta_{a,t}^{(k-1)})) \\
&= -2B_{as}^T(y_t - B_{as}\theta_{a,t}^{(k-1)} - \mu_t^{(k)})
\end{aligned}$$

The last equation holds because of (4.9).

$$\begin{aligned}
\theta_{a,t}^{(k)} &= \underset{\theta_{a,t}}{\operatorname{argmin}}\{\left\langle \theta_{a,t} - \theta_{a,t}^{(k-1)}, \nabla f(\theta_{a,t}^{(k-1)}) \right\rangle + \frac{L}{2}\|\theta_{a,t} - \theta_{a,t}^{(k-1)}\|^2 + \gamma\|\theta_{a,t}\|_1\} \\
&= \underset{\theta_{a,t}}{\operatorname{argmin}} \frac{L}{2}\|\theta_{a,t} - \theta_{a,t}^{(k-1)} - \frac{2}{L}B_{as}^T(y_t - B_{as}\theta_{a,t}^{(k-1)} - \mu_t^{(k)})\|^2 + \gamma\|\theta_{a,t}\|_1\}
\end{aligned}$$

We know that this can be solved by the soft-thresholding operator as follow: $\theta_{a,t}^{(k)} = S_{\frac{\gamma}{L}}(\theta_{a,t}^{(k-1)} + \frac{2}{L}B_{as}^T(y - B_{as}\theta_{a,t}^{(k-1)} - \mu_t^{(k)}))$ which is exactly (A.3).

Appendix H: The limit of the temporal projection matrix for the static background

Proposition. *The temporal projection matrix* H_t *in* (4.7) *and* (4.5) *becomes the average projection matrix* $H_t \to \frac{1}{n}1_n 1_n^T$ *when* $\lambda_t \to \infty$ *and* $c \to 0$*, respectively, where* $1_n = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ *is the column vector of* 1.

Proof. To prove this, we look at the following two lemmas

Lemma 1: For the roughness minimization model: $H_t = (I + \lambda_t D_t^T D_t)^{-1} \to \frac{1}{n} 1_n 1_n^T$ **when** $\lambda \to \infty$**.** Suppose the eigendecomposition of $D^T D$ yields, $D^T D = U \Lambda U^{-1}$. It has been proved in that $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_n)$ with $\lambda_i = -2 + 2\cos((i-1)\pi/n)$. This gives that there is only one eigenvalue that equals to 0 as $\lambda_1 = 0$, and $\lambda_i \neq 0$, when $i \geq 2$.

$(I + \lambda\Lambda)^{-1} = \text{diag}(\frac{1}{1+\lambda\lambda_1}, \cdots, \frac{1}{1+\lambda\lambda_n}) \to \text{diag}(1, 0 \cdots 0)$

$$
\begin{aligned}
H_t &= (I + \lambda_t U \Lambda U^{-1})^{-1} = U^T (I + \lambda_t \Lambda)^{-1} U \\
&\to U^T \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} U = u u^T
\end{aligned}
$$

, in which u is the first eigenvector of $D^T D$, which corresponds to eigenvalue 0. It is not hard to show that $u = \frac{1}{\sqrt{n}} 1_n$, in which $1_n = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ because $D^T D u = \frac{1}{\sqrt{n}} D^T D 1_n = 0$. Therefore $H = \frac{1}{\sqrt{n}} 1_n \frac{1}{\sqrt{n}} 1_n^T = \frac{1}{n} 1_n 1_n^T$

Lemma 2: For the kernel model: $H_t = K_t(K_t + \lambda_t I)^{-1} \propto 1_n 1_n^T$ **when** $c \to 0$**.** When $c \to \infty$, sine $\kappa(i,j) = \exp(-\frac{(i-j)^2}{2c^2})$,

Therefore, $K_t = 1_n 1_n^T$ and

$$H_t = K_t(K_t + \lambda_t I)^{-1}$$

From ShermanMorrison formula we know that

$$
\begin{aligned}
(K_t + \lambda_t I)^{-1} &= (1_n 1_n^T + \lambda_t I)^{-1} = \frac{1}{\lambda_t}(I + \frac{1}{\lambda_t} 1_n 1_n^T)^{-1} \\
&= \frac{1}{\lambda_t}(I - \frac{1}{\lambda_t}\frac{1_n 1_n^T}{1 + 1_n^T 1_n}) = \frac{1}{\lambda_t}(I - \frac{1}{\lambda_t(n+1)} 1_n 1_n^T)
\end{aligned}
$$

This gives

$$
H = K_t(K_t + \lambda_t I)^{-1} = 1_n 1_n^T \frac{1}{\lambda_t}(I - \frac{1}{\lambda_t(n+1)} 1_n 1_n^T) \propto 1_n 1_n^T
$$

□

APPENDIX C

APPENDIX ON "AN ADAPTIVE FRAMEWORK FOR ONLINE SENSING AND ANOMALY DETECTION "

Appendix A: The Proof of Proposition (9)

Proof. The function $g_a(r) = (K_h(r, r_a) + u)\|r - r_a\|^\lambda$ only relies on the distance $\|r - r_a\|$ since the gaussian kernel $K_h(r, r_a) = \frac{1}{(\sqrt{2\pi}h)^2}\exp(-\frac{\|r-r_a\|^2}{2h^2})$ can be represented as a function of $\|r - r_a\|$. Let us define $d := \|r - r_a\|$. Therefore, $g_a(r) = \tilde{g}_a(d) = (p_a\frac{1}{2\pi h^2}\exp(-\frac{d^2}{2h^2}) + u)d^\lambda$. The local optimum can be obtained by solving $\tilde{g}'_a(d) = 0$, which is

$$\tilde{g}'_a(d) = \frac{1}{2\pi h^4}\exp(-\frac{d^2}{2h^2})d^{\lambda-1}(2\pi h^4\lambda u\exp(\frac{d^2}{2h^2}) - p_a d^2 + h^2\lambda p_a) = 0.$$

Consequently, $2\pi h^2\frac{\lambda u}{p_a}\exp(\frac{d^2}{2h^2}) + \lambda = \frac{d^2}{h^2}$, which is equivalent to solving

$$-\frac{\pi h^2\lambda u}{p_a}\exp(\frac{\lambda}{2}) = (-\frac{d^2}{2h^2} + \frac{\lambda}{2})\exp(-\frac{d^2}{2h^2} + \frac{\lambda}{2}).$$

The above equation can be solved analytically by Lambert W-function $-\frac{d^2}{2h^2}+\frac{\lambda}{2} = W(-\frac{\pi h^2\lambda u}{p_a}\exp(\frac{\lambda}{2}))$. After some simplification, we have

$$d_a^* = h\sqrt{\lambda - 2W(-\frac{\pi h^2\lambda u}{p_a}\exp(\frac{\lambda}{2}))}.$$

□

Appendix B: The Proof of Proposition (10)

Proof. We first consider the case that $g(r)$ is in the neighborhood of r_a, defined as $\mathcal{R}_a = \{r | \|r - r_a\| \leq \|r - r_k\|, \forall k = 1, \cdots, n\}$. Therefore,

$$\max_{r \in \mathcal{R}_a} g(r) = \max_{r \in \mathcal{R}_a} g_a(r) = g_a(d_a^*) = \frac{u(d_a^*)^{\lambda+2}}{(d_a^*)^2 - \lambda h^2}$$

We then consider $g(r)$ in the neighborhood of other points r_j, which $r_j \neq r_a$.

$$\begin{aligned} \max_{r \in \mathcal{R}_j, r_j \neq r_a} g(r) &= (\frac{p_a}{2\pi h^2} \exp(-\frac{\|r - r_a\|^2}{2h^2}) + u)\|r - r_j\|^\lambda \\ &\leq u(1 + \exp(-c^2))d^\lambda \end{aligned}$$

The last inequality holds since $\|r - r_a\|_{r \in R_j} \geq \frac{1}{2}\|r_j - r_a\| \geq c\sqrt{2h^2 \ln(\frac{p_a}{2\pi h^2 u})}$, which means, $\frac{p_a}{2\pi h^2} \exp(-\frac{\|r-r_a\|^2}{2h^2}) \leq u \exp(-c^2)$. If

$$d < d_a^*(\frac{(d_a^*)^2}{2((d_a^*)^2 - \lambda h^2)})^{\frac{1}{\lambda}}(\frac{1}{(1 + \exp(-c^2)}})^{\frac{1}{\lambda}},$$

then, $\max_{r \in \mathcal{R}_j, j \neq a} g(r) \leq \max_{r \in \mathcal{R}_a} g(r)$. Therefore, $\operatorname{argmax}_r g(r)$ can be found in the neighborhood of r_a. More specifically, according to in Proposition 1, $\operatorname{argmax}_r g_a(r) = \{r | \|r - r_a\| = d_a^*\}$. □

Appendix C: Iterative Soft-thresholding for Robust Kernel Regression

We first show the equivalency of robust kernel regression and outlier detection in the following Lemma.

Lemma. (C.1) *and* (5.4) *are equivalent in the sense that the* μ *solved by both formulations are the same.*

$$\min_{a,\mu} \|z - a - \mu\|_2^2 + \gamma\|a\|_1 + \lambda\|\mu\|_H \tag{C.1}$$

The detailed proof of this is shown in [117].

It is straightforward to show that if μ is given, a can be solved by soft-thresholding as $a = S(z - \mu, \frac{\gamma}{2})$, where $S(x, \frac{\gamma}{2}) = \mathrm{sgn}(x)(|x| - \frac{\gamma}{2})_+$ is the soft-thresholding operator. $\mathrm{sgn}(x)$ is the sign function and $x_+ = \max(x, 0)$. This Lemma relates the robust kernel regression with outlier detection problem, which also explains why $\frac{\gamma}{2}$ is a natural threshold for a point to be considered as an outlier. Furthermore, given a, μ can be solved via $\mu = H(z - a)$, where H is the projection matrix computed by $H = K(K + \lambda I)^{-1}$.

Algorithm 10: Optimization algorithm for Robust Kernel Regression

initialize
- Choose a basis for the background as B
- $a^{(0)} = 0$
- $H = K(K + \lambda I)^{-1}$

end

while $|\mu^{(t-1)} - \mu^{(t)}| > \epsilon$ **do**
- Update $\mu^{(t+1)}$ via $\mu^{(t+1)} = H(z - a^{(t)})$
- Update $a^{(t+1)}$ by $a^{(t+1)} = S(z - \mu^{(t+1)}, \frac{\gamma}{2}))$

end

APPENDIX D

APPENDIX ON "POINT CLOUD DATA MODELING AND ANALYSIS VIA REGULARIZED TENSOR REGRESSION AND DECOMPOSITION"

Appendix A: The proof of Proposition 11

Proof. $\mathcal{B}$ can be solved by

$$\begin{aligned}\hat{\mathcal{B}} &= \underset{\mathcal{B}}{\operatorname{argmin}} \|\mathcal{Y} - \mathcal{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{X}\|_F^2 \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{y} - (\mathbf{X} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})\boldsymbol{\beta}\|_F^2\end{aligned}$$

Where $\tilde{\mathbf{y}}$ is the vectorized $\tilde{\mathcal{Y}}$ with size $\mathbf{y} \in \mathbb{R}^{\mathrm{n_1 n_2 N} \times 1}$, $\hat{\boldsymbol{\beta}}$ is the vectorized $\hat{\mathcal{B}}$ with size $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{pI_1I_2 \times 1}$, which can be solved efficiently by

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= ((\mathbf{X} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})^T(\mathbf{X} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)}))^{-1}(\mathbf{X} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})^T\mathbf{y} \\ &= (\mathbf{X}^T\mathbf{X})^{-1} \otimes (\mathbf{U}^{(2)^T}\mathbf{U}^{(2)})^{-1} \otimes (\mathbf{U}^{(1)^T}\mathbf{U}^{(1)})^{-1})(\mathbf{X} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})^T\mathbf{y} \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \otimes (\mathbf{U}^{(2)^T}\mathbf{U}^{(2)})^{-1}\mathbf{U}^{(2)^T} \otimes (\mathbf{U}^{(1)^T}\mathbf{U}^{(1)})^{-1}\mathbf{U}^{(1)^T}\mathbf{y}\end{aligned}$$

Or equivalently

$$\begin{aligned}\hat{\mathcal{B}} &= \tilde{\mathcal{Y}} \times_1 (\mathbf{U}^{(1)^T}\mathbf{U}^{(1)})^{-1}\mathbf{U}^{(1)^T} \times_2 (\mathbf{U}^{(2)^T}\mathbf{U}^{(2)})^{-1}\mathbf{U}^{(2)^T} \times_3 (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \\ &= \hat{\mathcal{S}} \times_1 (\mathbf{U}^{(1)^T}\mathbf{U}^{(1)})^{-1} \times_2 (\mathbf{U}^{(2)^T}\mathbf{U}^{(2)})^{-1} \times_3 (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\end{aligned}$$

Notice that which is exactly the same as computing $\hat{\mathcal{S}}$ first in 6.8 and then perform regression on $\hat{\mathcal{S}}$ given the input variable $\mathbf{X}$. □

Appendix B: The proof of Proposition 12

Proof. By rewriting the objective function $\|\mathcal{Y} - \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}\|_F^2 + s^T\mathbf{R}s$ into $\|y - \mathbf{U}s\|^2 + s^T\mathbf{R}s$ where $\mathbf{U} = \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)}$, $s = \text{vec}(\mathcal{S})$, and $y = \text{vec}(\mathcal{Y})$. This can be solved by

$$\begin{aligned} s &= (\mathbf{U}^T\mathbf{U} + \mathbf{R})^{-1}\mathbf{U}^T y \\ &= (\mathbf{U}_2^T(\mathbf{I}_{I_2} + \lambda(\mathbf{D}_2^2)^T\mathbf{D}_2^2)\mathbf{U}_2)^{-1} \otimes (\mathbf{U}_1^T(\mathbf{I}_{I_1} + \lambda(\mathbf{D}_1^2)^T\mathbf{D}_1^2)\mathbf{U}_1)^{-1}y \\ &= \mathbf{U}^T y \end{aligned}$$

The last equation holds since we constrain $\mathbf{U}_k^T(\mathbf{I}_{I_k} + \lambda(\mathbf{D}_k^2)^T\mathbf{D}_k^2)\mathbf{U}_k = \mathbf{I}_{P_k}$, we can then plug s into (6.13), we have

$$\begin{aligned} \|y - \mathbf{U}s\|^2 + s^T\mathbf{R}s = s^T(\mathbf{U}^T\mathbf{U} + \mathbf{R})s - 2s^T\mathbf{U}^T y = -y^T\mathbf{U}\mathbf{U}^T y \\ = -\|\mathbf{U}^T y\|_F^2 = -\|\mathcal{Y} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T\|_F^2 \end{aligned}$$

Therefore, minimizing (6.13) is equivalent to maximize the (6.12) under the constrain $\mathbf{U}_k^T(\mathbf{I}_{I_k} + \lambda(\mathbf{D}_k^2)^T\mathbf{D}_k^2)\mathbf{U}_k = \mathbf{I}_{P_k}$.

□

Appendix C: The proof of Proposition 13

Proof. Given $\mathbf{U}^{(-k)}$, the solution of $\text{argmax}_{\mathbf{U}^{(k)}} \|\mathcal{Y} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T}\|_F^2$ is given by the following generalized eigenvalue problem

$$\mathbf{W}_k\mathbf{U}^{(k)} = \mathbf{R}_k(\lambda)\mathbf{U}^{(k)}\Lambda_k$$

where $\mathbf{W}_k = \mathbf{Y}_{(k)}\mathbf{U}^{(-k)}\mathbf{U}^{(-k)^T}\mathbf{Y}_{(k)}^T$, Λ_k is the diagonal eigenvalue matrix, $\mathbf{U}^{(-k)} =$

$$\begin{cases} \mathbf{U}^{(2)} & k=1 \\ \mathbf{U}^{(1)} & k=2 \end{cases}.$$

First, $\mathcal{Z} = \mathcal{Y} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T}$, $Z_{(1)} = \mathbf{U}^{(1)^T}\mathbf{Y}_{(1)}\mathbf{U}^{(2)}$, $Z_{(2)} = \mathbf{U}^{(2)^T}\mathbf{Y}_{(2)}\mathbf{U}^{(1)}$

$$\begin{aligned} \|\mathcal{Y} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T}\|_F^2 &= \|\mathbf{U}^{(k)^T}\mathbf{Y}_{(k)}\mathbf{U}^{(-k)}\|_F^2 = tr(\mathbf{U}^{(k)^T}\mathbf{Y}_{(k)}\mathbf{U}^{(-k)}\mathbf{U}^{(-k)^T}\mathbf{Y}_{(k)}^T\mathbf{U}^{(k)}) \\ &= tr(\mathbf{U}^{(k)^T}\mathbf{W}_k\mathbf{U}^{(k)}) \end{aligned}$$

where $\mathbf{W}_k = \mathbf{Y}_{(k)}\mathbf{U}^{(-k)}\mathbf{U}^{(-k)^T}\mathbf{Y}_{(k)}^T$. Therefore, $\hat{\mathbf{U}}^{(k)} = \text{argmin}_{\mathbf{U}^{(k)}} tr(\mathbf{U}^{(k)^T}\mathbf{W}_k\mathbf{U}^{(k)})$ can be solved by generalized eigenvalue problem.

$$\mathbf{W}_k\mathbf{U}^{(k)} = \mathbf{R}_k(\lambda)\mathbf{U}^{(k)}\Lambda_k$$

□

Appendix D: The proof of Proposition 13

Proof. $\boldsymbol{\beta}$ can be solved by

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \underset{\boldsymbol{\beta}}{argmin} \|\mathbf{y} - (\mathbf{X} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})\boldsymbol{\beta}\|^2 + P(\boldsymbol{\beta}) \\ &= \underset{\boldsymbol{\beta}}{argmin}\, \boldsymbol{\beta}^T(\mathbf{X}^T\mathbf{X}) \otimes \mathbf{U}^{(2)^T}\mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)^T}\mathbf{U}^{(1)}\boldsymbol{\beta} + P(\boldsymbol{\beta}) - 2\boldsymbol{\beta}^T(\mathbf{X}^T \otimes \mathbf{U}^{(2)^T} \otimes \mathbf{U}^{(1)^T})\mathbf{y} \\ &= \underset{\boldsymbol{\beta}}{argmin}\, \boldsymbol{\beta}^T(\mathbf{X}^T\mathbf{X}) \otimes (\mathbf{U}^{(2)^T}\mathbf{U}^{(2)} + \lambda\mathbf{P}_2) \otimes (\mathbf{U}^{(1)^T}\mathbf{U}^{(1)} + \lambda\mathbf{P}_1) - 2\boldsymbol{\beta}^T(\mathbf{X}^T \otimes \mathbf{U}^{(2)^T} \otimes \mathbf{U}^{(1)^T})\mathbf{y} \\ &= (\mathbf{X}^T\mathbf{X})^{-1} \otimes (\mathbf{U}^{(2)^T}\mathbf{U}^{(2)} + \lambda\mathbf{P}_2)^{-1} \otimes (\mathbf{U}^{(1)^T}\mathbf{U}^{(1)} + \lambda\mathbf{P}_1)^{-1}(\mathbf{X}^T \otimes \mathbf{U}^{(2)^T} \otimes \mathbf{U}^{(1)^T})\mathbf{y} \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \otimes (\mathbf{U}^{(2)^T}\mathbf{U}^{(2)} + \lambda\mathbf{P}_2)^{-1}\mathbf{U}^{(2)^T} \otimes (\mathbf{U}^{(1)^T}\mathbf{U}^{(1)} + \lambda\mathbf{P}_1)^{-1}\mathbf{U}^{(1)^T}\mathbf{y} \end{aligned}$$

which is equivalent to solve $\mathcal{B}$ in the tensor format as shown in (6.16). □

REFERENCES

[1] Y. Lei, Z. Zhang, and J. Jin, "Automatic tonnage monitoring for missing part detection in multi-operation forging processes," *Journal of manufacturing science and engineering*, vol. 132, no. 5, p. 051 010, 2010.

[2] "Identification of influential functional process variables for surface quality control in hot rolling processes," *Ieee transactions on automation science and engineering*, vol. 5, no. 3, pp. 557–562, Jul. 2008.

[3] H. Yan, K. Paynabar, and J. Shi, "Image-based process monitoring using low-rank tensor decomposition," *Ieee transactions on automation science and engineering*, vol. 12, no. 1, pp. 216–227, Jan. 2015.

[4] E. A. Patterson and Z. F. Wang, "Towards full field automated photoelastic analysis of complex components," *Strain*, vol. 27, no. 2, pp. 49–53, May 1991.

[5] H. Sohn, G. Park, J. R. Wait, N. P. Limback, and C. R. Farrar, "Wavelet-based active sensing for delamination detection in composite structures," *Smart materials and structures*, vol. 13, no. 1, pp. 153–160, Dec. 2003.

[6] B. C. Jiang *, C.-C. Wang, and H.-C. Liu, "Liquid crystal display surface uniformity defect inspection using analysis of variance and exponentially weighted moving average techniques," *International journal of production research*, vol. 43, no. 1, pp. 67–80, Jan. 2005.

[7] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *Ieee transactions on industrial electronics*, vol. 55, no. 1, pp. 348–363, Jan. 2008.

[8] *Structural health monitoring*. Wiley-Blackwell, Jan. 2006.

[9] H. Yan, K. Paynabar, and J. Shi, "Anomaly detection in images with smooth background via smooth-sparse decomposition," *Technometrics*, vol. 59, no. 1, pp. 102–114, Jan. 2017.

[10] J. A. Simpson, "Mechanical measurement and manufacturing," *Control and dynamic systems: Advances in theory and applications*, vol. 45, p. 17, 1992.

[11] O. Mesnil, C. A. C. Leckey, and M. Ruzzene, "Instantaneous wavenumber estimation for damage quantification in layered plate structures," in *Health monitoring of structural and biological systems 2014*, SPIE-Intl Soc Optical Eng, Mar. 2014.

[12] J.-D. Aussel and J.-P. Monchalin, "Precision laser-ultrasonic velocity measurement and elastic constant determination," *Ultrasonics*, vol. 27, no. 3, pp. 165–177, May 1989.

[13] R. Jin, C.-J. Chang, and J. Shi, "Sequential measurement strategy for wafer geometric profile estimation," *Iie transactions*, vol. 44, no. 1, pp. 1–12, Jan. 2012.

[14] O. Mesnil and M. Ruzzene, "Sparse wavefield reconstruction and source detection using compressed sensing," *Ultrasonics*, vol. 67, pp. 94–104, Apr. 2016.

[15] L. Devroye and A. Krzyżak, "On the hilbert kernel density estimate," *Statistics probability letters*, vol. 44, no. 3, pp. 299–308, Sep. 1999.

[16] I. Gibson, D. W. Rosen, and B. Stucker, *Additive manufacturing technologies*. Springer Nature, 2010.

[17] Q. Huang, J. Zhang, A. Sabbaghi, and T. Dasgupta, "Optimal offline compensation of shape shrinkage for three-dimensional printing processes," *Iie transactions*, vol. 47, no. 5, pp. 431–441, Oct. 2014.

[18] M. Levoy, J. Ginsberg, J. Shade, D. Fulk, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, and J. Davis, "The digital michelangelo project," in *Proceedings of the 27th annual conference on computer graphics and interactive techniques - siggraph '00*, Association for Computing Machinery (ACM), 2000.

[19] E. Huising and L. Gomes Pereira, "Errors and accuracy estimates of laser data acquired by various laser scanning systems for topographic applications," *Isprs journal of photogrammetry and remote sensing*, vol. 53, no. 5, pp. 245–261, Oct. 1998.

[20] H. Schwenke, W. Knapp, H. Haitjema, A. Weckenmann, R. Schmitt, and F. Delbressine, "Geometric error measurement and compensation of machines—an update," *Cirp annals - manufacturing technology*, vol. 57, no. 2, pp. 660–675, 2008.

[21] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 ieee international conference on robotics and automation*, Institute of Electrical and Electronics Engineers (IEEE), May 2011.

[22] M. Pieraccini, G. Guidi, and C. Atzeni, "3d digitizing of cultural heritage," *Journal of cultural heritage*, vol. 2, no. 1, pp. 63–70, Jan. 2001.

[23] C. de Boor, *A practical guide to splines*. Springer New York, 1978.

[24] A. Wink and J. Roerdink, "Denoising functional mr images: A comparison of wavelet denoising and gaussian smoothing," *Ieee transactions on medical imaging*, vol. 23, no. 3, pp. 374–387, Mar. 2004.

[25] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Computer graphics and image processing*, vol. 9, no. 4, pp. 394–407, Apr. 1979.

[26] P. Qiu, "Jump surface estimation, edge detection, and image restoration," *Journal of the american statistical association*, vol. 102, no. 478, pp. 745–756, Jun. 2007.

[27] P. Qiu and P. S. Mukherjee, "Edge structure preserving 3-D image denoising by local surface approximation," *Pattern analysis and machine intelligence, ieee transactions on*, vol. 34, no. 8, pp. 1457–1468, 2012.

[28] M. Unser, "Splines: A perfect fit for signal and image processing," *Ieee signal processing magazine*, vol. 16, no. 6, pp. 22–38, 1999.

[29] P. H. C. Eilers and B. D. Marx, "Flexible smoothing with b -splines and penalties," *Statistical science*, vol. 11, no. 2, pp. 89–121, May 1996.

[30] M. P. Wand and M. C. Jones, *Kernel smoothing*. Springer Nature, 1995.

[31] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *Ieee signal processing magazine*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[32] A. Kumar and G. Pang, "Defect detection in textured materials using gabor filters," *Ieee transactions on industry applications*, vol. 38, no. 2, pp. 425–440, 2002.

[33] C.-J. Lu and D.-M. Tsai *, "Defect inspection of patterned thin film transistor-liquid crystal display panels using a fast sub-image-based singular value decomposition," *International journal of production research*, vol. 42, no. 20, pp. 4331–4351, Oct. 2004.

[34] "Fabric defect detection by fourier analysis," *Ieee transactions on industry applications*, vol. 36, no. 5, pp. 1267–1276, 2000.

[35] J. Funck, Y Zhong, D. Butler, C. Brunner, and J. Forrer, "Image segmentation algorithms applied to wood defect detection," *Computers and electronics in agriculture*, vol. 41, no. 1-3, pp. 157–179, Dec. 2003.

[36] I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing," 1968.

[37] J. M. Prewitt, "Object enhancement and extraction," *Picture processing and psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970.

[38] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the royal society b: Biological sciences*, vol. 207, no. 1167, pp. 187–217, Feb. 1980.

[39] P. Qiu and B. Yandell, "Jump detection in regression surfaces," *Journal of computational and graphical statistics*, vol. 6, no. 3, p. 332, Sep. 1997.

[40] P. Qiu and J. Sun, "Using conventional edge detectors and postsmoothing for segmentation of spotted microarray images," *Journal of computational and graphical statistics*, vol. 18, no. 1, pp. 147–164, Jan. 2009.

[41] ——, "Local smoothing image segmentation for spotted microarray images," *Journal of the american statistical association*, vol. 102, no. 480, pp. 1129–1144, Dec. 2007.

[42] C. Park, J. Z. Huang, D. Huitink, S. Kundu, B. K. Mallick, H. Liang, and Y. Ding, "A multistage, semi-automated procedure for analyzing the morphology of nanoparticles," *Iie transactions*, vol. 44, no. 7, pp. 507–522, Jul. 2012.

[43] N. Otsu, "A threshold selection method from gray-level histograms," *Ieee transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[44] "Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images," *Ieee transactions on pattern analysis and machine intelligence*, vol. 25, no. 1, pp. 131–137, Jan. 2003.

[45] D. Bradley and G. Roth, "Adaptive thresholding using the integral image," *Journal of graphics tools*, vol. 12, no. 2, pp. 13–21, Jan. 2007.

[46] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern recognition*, vol. 33, no. 2, pp. 225–236, Feb. 2000.

[47] R. Adams and L. Bischof, "Seeded region growing," *Ieee transactions on pattern analysis and machine intelligence*, vol. 16, no. 6, pp. 641–647, Jun. 1994.

[48] P. Soille, *Morphological image analysis*. Springer Nature, 1999.

[49] Y. Mei, "Efficient scalable schemes for monitoring a large number of data streams," *Biometrika*, vol. 97, no. 2, pp. 419–433, Apr. 2010.

[50] K. Liu, Y. Mei, and J. Shi, "An adaptive sampling strategy for online high-dimensional process monitoring," *Technometrics*, vol. 57, no. 3, pp. 305–319, Aug. 2014.

[51] C. Zou, Z. Wang, X. Zi, and W. Jiang, "An efficient online monitoring method for high-dimensional data streams," *Technometrics*, vol. 57, no. 3, pp. 374–387, Jul. 2014.

[52] P. Qiu and D. Xiang, "Univariate dynamic screening system: An approach for identifying individuals with irregular longitudinal behavior," *Technometrics*, vol. 56, no. 2, pp. 248–260, Apr. 2014.

[53] D. Xiang, P. Qiu, and X. Pu, "Nonparametric regression analysis of multivariate longitudinal data," *Statistica sinica*, 2013.

[54] C. Zou, F. Tsung, and Z. Wang, "Monitoring profiles based on nonparametric regression methods," *Technometrics*, vol. 50, no. 4, pp. 512–526, Nov. 2008.

[55] P. Qiu, C. Zou, and Z. Wang, "Nonparametric profile monitoring by mixed effects modeling," *Technometrics*, vol. 52, no. 3, pp. 265–277, Aug. 2010.

[56] S. I. Chang and S. Yadama, "Statistical process control for monitoring non-linear profiles using wavelet filtering and b-spline approximation," *International journal of production research*, vol. 48, no. 4, pp. 1049–1068, Feb. 2010.

[57] K. Paynabar and J. J. Jin, "Characterization of non-linear profiles variations using mixed-effect models and wavelets," *Iie transactions*, vol. 43, no. 4, pp. 275–290, Jan. 2011.

[58] R. Y. Liu, "Control charts for multivariate processes," *Journal of the american statistical association*, vol. 90, no. 432, pp. 1380–1387, Dec. 1995.

[59] K. Paynabar, C. Zou, and P. Qiu, "A change-point approach for phase-i analysis in multivariate profile monitoring and diagnosis," *Technometrics*, vol. 58, no. 2, pp. 191–204, Apr. 2016.

[60] B. R. Bakshi, "Multiscale pca with application to multivariate statistical process monitoring," *Aiche journal*, vol. 44, no. 7, pp. 1596–1610, Jul. 1998.

[61] J. L. Loeppky, L. M. Moore, and B. J. Williams, "Batch sequential designs for computer experiments," *Journal of statistical planning and inference*, vol. 140, no. 6, pp. 1452–1464, Jun. 2010.

[62] P. Ranjan, D. Bingham, and G. Michailidis, "Sequential experiment design for contour estimation from complex computer codes," *Technometrics*, vol. 50, no. 4, pp. 527–541, Nov. 2008.

[63] D. R. Jones, M. Schonlau, and W. J. Welch, *Journal of global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[64] P. de Aguiar, B. Bourguignon, M. Khots, D. Massart, and R. Phan-Than-Luu, "D-optimal designs," *Chemometrics and intelligent laboratory systems*, vol. 30, no. 2, pp. 199–210, Dec. 1995.

[65] F. Xiong, Y. Xiong, W. Chen, and S. Yang, "Optimizing latin hypercube design for sequential sampling of computer experiments," *Engineering optimization*, vol. 41, no. 8, pp. 793–810, Aug. 2009.

[66] P. C. Kyriakidis, "Sequential spatial simulation using latin hypercube sampling," in *Geostatistics banff 2004*, Springer Nature, 2005, pp. 65–74.

[67] E. Stinstra, D. den Hertog, P. Stehouwer, and A. Vestjens, "Constrained maximin designs for computer experiments," *Technometrics*, vol. 45, no. 4, pp. 340–346, Nov. 2003.

[68] J. L. Loeppky, J. Sacks, and W. J. Welch, "Choosing the sample size of a computer experiment: A practical guide," *Technometrics*, vol. 51, no. 4, pp. 366–376, Nov. 2009.

[69] V. R. Joseph, T. Dasgupta, R. Tuo, and C. F. J. Wu, "Sequential exploration of complex surfaces using minimum energy designs," *Technometrics*, vol. 57, no. 1, pp. 64–74, Jan. 2015.

[70] J. Zhu, S. C. H. Hoi, and M. R.-T. Lyu, "Robust regularized kernel regression," *Ieee transactions on systems, man, and cybernetics, part b (cybernetics)*, 2009.

[71] C. de Boor, "On calculating with b-splines," *Journal of approximation theory*, vol. 6, no. 1, pp. 50–62, Jul. 1972.

[72] H. Yan, K. Paynabar, and J. Shi, "Anomaly detection in images with smooth background via smooth-sparse decomposition," *Technometrics*, vol. 59, no. 1, pp. 102–114, Jan. 2017.

[73] B. Baumgart, *Winged-edge polyhedron representation for computer vision. national computer conference, may 1975*, 1975.

[74] *An introduction to nurbs*. Elsevier BV, 2001.

[75] J. R. Shewchuk, "Triangle: Engineering a 2d quality mesh generator and delaunay triangulator," in *Applied computational geometry towards geometric engineering*, Springer Nature, 1996, pp. 203–222.

[76] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *Acm transactions on graphics*, vol. 13, no. 1, pp. 43–72, Jan. 1994.

[77] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *Ieee transactions on visualization and computer graphics*, vol. 5, no. 4, pp. 349–359, Oct. 1999.

[78] B. M. Colosimo, F. Mammarella, and S. Petrò, "Quality control of manufactured surfaces," in *Frontiers in statistical quality control 9*, Springer Nature, 2010, pp. 55–70.

[79] L. J. Wells, F. M. Megahed, C. B. Niziolek, J. A. Camelio, and W. H. Woodall, "Statistical process monitoring approach for high-density point clouds," *Journal of intelligent manufacturing*, vol. 24, no. 6, pp. 1267–1279, Jun. 2012.

[80] B. M. Colosimo, P. Cicorella, M. Pacella, and M. Blaco, "From profile to surface monitoring: Spc for cylindrical surfaces via gaussian processes," *Journal of quality technology*, vol. 46, no. 2, p. 95, 2014.

[81] I. T. Jolliffe, *Principal component analysis*. Springer New York, 1986.

[82] P. T. Reiss, L. Huang, and M. Mennes, "Fast function-on-scalar regression with penalized basis expansions," *The international journal of biostatistics*, vol. 6, no. 1, Jan. 2010.

[83] I. S. Helland, "Partial least squares regression and statistical models," *Scandinavian journal of statistics*, pp. 97–114, 1990.

[84] J. Peng, J. Zhu, A. Bergamaschi, W. Han, D.-Y. Noh, J. R. Pollack, and P. Wang, "Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer," *The annals of applied statistics*, vol. 4, no. 1, pp. 53–77, Mar. 2010.

[85] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *Journal of the american statistical association*, vol. 108, no. 502, pp. 540–552, Jun. 2013.

[86] *Statistical parametric mapping*. Elsevier BV, 2007.

[87] Y. Li, H. Zhu, D. Shen, W. Lin, J. H. Gilmore, and J. G. Ibrahim, "Multiscale adaptive regression models for neuroimaging data," *Journal of the royal statistical society: Series b (statistical methodology)*, vol. 73, no. 4, pp. 559–578, Mar. 2011.

[88] L. Li and X. Zhang, "Parsimonious tensor response regression," *Journal of the american statistical association*, pp. 0–0, Jun. 2016.

[89] R. Tibshirani, "Regression shrinkage and selection via the lasso: A retrospective," *Journal of the royal statistical society: Series b (statistical methodology)*, vol. 73, no. 3, pp. 273–282, Apr. 2011.

[90] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*. Society for Industrial Applied Mathematics (SIAM), Jan. 1994.

[91] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *The annals of applied statistics*, vol. 1, no. 2, pp. 302–332, Dec. 2007.

[92] R. Tibshirani, I. Johnstone, T. Hastie, and B. Efron, "Least angle regression," *The annals of statistics*, vol. 32, no. 2, pp. 407–499, Apr. 2004.

[93] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *Submitted to siam journal on optimization*, 2008.

[94] N. Parikh, "Proximal algorithms," *Foundations and trends® in optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[95] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," in *Soviet mathematics doklady*, vol. 27, 1983, pp. 372–376.

[96] L. Xiao, Y. Li, and D. Ruppert, "Fast bivariatep-splines: The sandwich smoother," *Journal of the royal statistical society: Series b (statistical methodology)*, vol. 75, no. 3, pp. 577–599, Feb. 2013.

[97] D. Ruppert, "Selecting the number of knots for penalized splines," *Journal of computational graphical statistics*, vol. 11, no. 4, pp. 735–757, Dec. 2002.

[98] J. Sun and P. Qiu, "Jump detection in regression surfaces using both first-order and second-order derivatives," *Journal of computational and graphical statistics*, vol. 16, no. 2, pp. 289–311, Jun. 2007.

[99] K. Khurshid, I. Siddiqi, C. Faure, and N. Vincent, "Comparison of niblack inspired binarization methods for ancient documents," in *Document recognition and retrieval xvi*, SPIE-Intl Soc Optical Eng, Jan. 2009.

[100] D. Garcia, "Robust smoothing of gridded data in one and higher dimensions with missing values," *Computational statistics data analysis*, vol. 54, no. 4, pp. 1167–1178, Apr. 2010.

[101] R. G. R. Prasath, K. Skenes, and S. Danyluk, "Comparison of phase shifting techniques for measuring in-plane residual stress in thin, flat silicon wafers," *Journal of electronic materials*, vol. 42, no. 8, pp. 2478–2485, Jun. 2013.

[102] M. Ramji and R. Prasath, "Sensitivity of isoclinic data using various phase shifting techniques in digital photoelasticity towards generalized error sources," *Optics and lasers in engineering*, vol. 49, no. 9-10, pp. 1153–1167, Sep. 2011.

[103] K. Skenes, G. Prasath, and S. Danyluk, "Silicon grain crystallographic orientation measurement from nir transmission and reflection," in *2013 ieee 39th photovoltaic specialists conference (pvsc)*, Institute of Electrical and Electronics Engineers (IEEE), Jun. 2013.

[104] D. Ruppert, "Selecting the number of knots for penalized splines," *Journal of computational graphical statistics*, vol. 11, no. 4, pp. 735–757, Dec. 2002.

[105] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel hilbert spaces in probability and statistics*. Springer Nature, 2004.

[106] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *International conference on computational learning theory*, Springer, 2001, pp. 416–426.

[107] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the gaussian kernel for scale-space filtering," *Ieee transactions on pattern analysis and machine intelligence*, vol. PAMI-8, no. 1, pp. 26–33, Jan. 1986.

[108] D. M. Hawkins, "Regression adjustment for variables in multivariate quality control," *Journal of quality technology*, vol. 25, no. 3, pp. 170–182, 1993.

[109] C. Zou and P. Qiu, "Multivariate statistical process control using lasso," *Journal of the american statistical association*, vol. 104, no. 488, pp. 1586–1596, Dec. 2009.

[110] S. Patankar, *Numerical heat transfer and fluid flow*. CRC Press, 1980.

[111] C. Zou, W. Jiang, and F. Tsung, "A lasso-based diagnostic framework for multivariate statistical process control," *Technometrics*, vol. 53, no. 3, pp. 297–309, Aug. 2011.

[112] S. Kalpakjian, S. R. Schmid, and C.-W. Kok, *Manufacturing processes for engineering materials*. Pearson-Prentice Hall, 2008.

[113] "An intelligent real-time vision system for surface defect detection," in *Proceedings of the 17th international conference on pattern recognition, 2004. icpr 2004.*, Institute of Electrical and Electronics Engineers (IEEE), 2004.

[114] Y. Xie, J. Huang, and R. Willett, "Change-point detection for high-dimensional time series with missing data," *Ieee journal of selected topics in signal processing*, vol. 7, no. 1, pp. 12–27, Feb. 2013.

[115] K. Paynabar, J. J. Jin, and M. Pacella, "Monitoring and diagnosis of multichannel nonlinear profile variations using uncorrelated multilinear principal component analysis," *Iie transactions*, vol. 45, no. 11, pp. 1235–1247, Nov. 2013.

[116] M. Johnson, L. Moore, and D. Ylvisaker, "Minimax and maximin distance designs," *Journal of statistical planning and inference*, vol. 26, no. 2, pp. 131–148, Oct. 1990.

[117] G. Mateos and G. B. Giannakis, "Robust nonparametric regression via sparsity control with application to load curve data cleansing," *Ieee transactions on signal processing*, vol. 60, no. 4, pp. 1571–1584, Apr. 2012.

[118] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sep. 1966.

[119] "Mpca: Multilinear principal component analysis of tensor objects," *Ieee transactions on neural networks*, vol. 19, no. 1, pp. 18–39, Jan. 2008.

[120] I. M. Johnstone and A. Y. Lu, "On consistency and sparsity for principal components analysis in high dimensions," *Journal of the american statistical association*, vol. 104, no. 486, pp. 682–693, Jun. 2009.

[121] B. W. Silverman, "Smoothed functional principal components analysis by choice of norm," *The annals of statistics*, vol. 24, no. 1, pp. 1–24, Feb. 1996.

[122] M. Pacella and B. M. Colosimo, "Multilinear principal component analysis for statistical modeling of cylindrical surfaces: A case study," *Quality technology quantitative management*, pp. 1–19, Sep. 2016.

[123] B. Western and D. Bloome, "9. variance function regressions for studying inequality," *Sociological methodology*, vol. 39, no. 1, pp. 293–326, Aug. 2009.

[124] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, Jun. 2001.

VITA

Hao Yan grew up in Tianjin, China. He received the B.S. Degree in Physics and secondary B.S. Degree in Economics from Peking University in 2011 and the M.S. degree in Statistics and Computational Science and Engineering from the Georgia Institute of Technology in 2015 and 2016.